Copyright
by
Sandeep Bhadra
2008

The Dissertation Committee for Sandeep Bhadra
certifies that this is the approved version of the following dissertation:

Network Coding for Next-Generation Networks

Committee:

Sanjay Shakkottai, Supervisor

Vijay K. Garg

Piyush Gupta

Gustavo de Veciana

Sriram Vishwanath

David Zuckerman

Network Coding for Next-Generation Networks

by

Sandeep Bhadra, B.Tech., M. Tech.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2008

To my parents

Acknowledgments

Working towards this dissertation has been a long and fruitful journey for me, personally and professionally; one in which, I have had the good fortune to interact with several remarkable people whom I wish to thank here.

Foremost among these, I would like to thank Sanjay Shakkottai, whose enthusiasm for theory is matched only by his keenness to relay that enthusiasm to his students. Sanjay has the remarkable quality of being able to make intuitive leaps at the initial stages of a problem and yet remain rigorous as our understanding of the problem matures. His counterexamples have frequently sent me back to my cubicle, clutching sheets torn from his stacks of yellow legal pads.

In my first few years of graduate school at UT, I benefited immensely from my interactions with Gustavo de Veciana and Sriram Vishwanath. From my early years at graduate school when I proposed my thesis topic, Gustavo has always urged me to keep an eye out for the big questions and to try to answer them as succinctly as I can; this approach has greatly influenced the structure of this thesis in terms of the choice of problems I tackle and the conclusions I seek to draw. Sriram has been an invaluable resource in my forays into network information theory; I have knocked on his door several times and have always come out with pointers to related work, together with his comments on how their results all fit together with each other, and with my problem.

When I began to consider network coding from a network design/analysis perspective, rather than an information theoretic perspective, I had the good fortune of collaborating closely with Piyush Gupta at Bell Labs. I would like to thank him for being supportive of my initial ideas on network information theory, for the countless hours of discussions in the process of refining them, and for helping me give shape to what are now Chapters 2 and 3 in this dissertation.

I would like to thank the other members of my thesis committee for their advice and encouragement. My conversations with Prof. Vijay Garg have helped me connect my problems to related topics in distributed algorithms and approximation theory, and to compare my work against theirs. Prof. David Zuckerman has patiently listened to my ideas; in my discussions with him during my thesis proposal and later at the defense, he pointed

out how my work on random linear source codes in Chapters 4 and 5 is closely related to early work on RS-codes and digital fountain codes over networks.

Outside UT, I have been lucky enough to work at really great places like Bell Labs, IBM Research and Texas Instruments (my current home). In addition to Piyush at Bell Labs, I would like to thank Gerhard Kramer and Phil Whiting for stimulating discussions on network information theory. I would like to thank Mark Squillante and Yingdong Lu for an exciting summer at IBM Research, Yorktown: learning stochastic optimization, and watching applied probability and business intersect in really interesting ways. Last, but not the least, I would like to thank Xiaolin Lu, Don Shaver and Martin Izzard for taking a chance on someone who did not write too many lines of code in graduate school and letting me work on a really interesting project on system design for 4G wireless systems in my summer at Texas Instruments. I am lucky enough to be able to continue working on the same project even now.

I would like to thank Profs. V.V. Rao and R. Aravind at IIT Madras, my alma mater, for getting me interested in communications and stochastic analysis; and for encouraging me to go to graduate school.

During the course of my stay at UT, I have kept great company at the Wireless Network and Communications Group. I have enjoyed working with my colleague Jung Ryu; much of the results of Chapter 4 have been in close collaboration with him. Moreover, Jung has been generous with his simulation results, which I presented in my defense to back the fixed-point bounds for TCP window size in that chapter. Chang-woo Yang, Shreeshankar Bodas and Sundar Subramanian have been ready references for my queries on real-analysis, queueing theory and sport. Brian Smith, who works on network information theory as well, has been kind enough to listen to my vague ideas and then proof-read some of my initial results. I thank him for our many discussions in the lab and beyond, and wish Sundar and him luck as they defend their theses soon.

As senior graduate students when I just stepped into graduate school, Manish Airy, Yung Yi, Wei Wu, Bishwarup Mondal and Antonio Forenza have been great peer mentors. Moreover, Manish, Bishwarup and Roopsha Samanta have been instrumental in ensuring that I eat healthy lunches at school. I thank Seung-Jun Baek, Siddhartha Bannerjee, Ramya Bhagavatule, Runhua Chen, Robert Daniels, Aditya Gopalan, Jared Grubb, Andrew Hunter, Kaibin Huang, Hongseok Kim, Caleb Lo, Marcel Nassar, Bilal Sadiq, Sriram Sridharan, Rahul Vaze and Marcus Young for making WNCG such an exciting place to be in. Most importantly, though, I must thank Janet Preuss and Melanie Gulick, for their humour and

good cheer, for their incredible patience, and for bailing me out of a pickle all too frequently. I apologize for anyone that I may have missed.

At home in Austin, I have had great roommates: Avinash Unnikrishnan first, and later my brother Sagar Bhadra. I thank them for their friendship and their ability to negotiate through the piles of "important" paper scattered in my apartment.

Finally, I would like to thank my family: my parents, Sanjib and Mala Bhadra, and my brother Sagar for their faith in me and for their love.

SANDEEP BHADRA
Austin, March 2008

Network Coding for Next-Generation Networks

Publication No. _____

Sandeep Bhadra, Ph.D.

The University of Texas at Austin, 2008

Supervisor: Sanjay Shakkottai

As a discipline at the intersection of information theory and classical stochastic network analysis, network coding promises interesting future applications, and hence presents newer fundamental theoretical problems in the field of network engineering and design. While much research on network coding is concerned with the analysis and construction of capacity achieving codes, our focus in this proposal will be on the impact of Random Linear Coding (RLC) in next generation wireline and wireless networks. We consider two techniques of coding for networks: one where coding is performed at every intermediate node of a network, and the other where only source nodes encode across packets. For either case, we present scenarios where network coding offers significant performance gains.

Under network coding at every intermediate node, the previously intractable min-cost multicast problem has been formulated in terms of a convex optimization. Recent work has focused on cooperative decentralized algorithms to solve this, most using primal-dual techniques. Instead, here we formulate a decentralized non-cooperative version of the problem where each user routes greedily to minimize its own cost and study how the resulting user-equilibrium cost compares to the global (social) optimum. Based on our results, simple greedy decentralized algorithms are proposed for distributed min- cost flow adaptation at nodes in the network.

In the context of wireless networks, achieving unicast capacity is complicated by wireless broadcast and interference. We note that while much of extant network coding research has been on wireline networks, our understanding of network codes applied to wireless networks is still limited. We abstract broadcast and interference properties in the wireless channel by a finite field addition channel, to arrive at a Broadcast and Additive

Interference Network (BAIN) and show that there exists a graph transformation, and a corresponding sample path coupling, to model a BAIN as a regular wireline network with network coding at intermediate nodes. Based on this analysis, we leverage existing results from network coding for wireline networks to arrive at asymptotically tight bounds on unicast capacity for BAINs.

Next, we consider network coding at the source, with no buffers at intermediate nodes, as an alternative to traditional buffering of transient packets at intermediate nodes in multi-hop networks, thereby virtually sharing memory between links on a flow path. We call this spatial buffer multiplexing: where buffering and coding implemented at the source alone compensates for packet loss at any downstream bufferless link. Using many-sources large deviations analysis, we show that network coding promises dramatic improvements in resource allocation and buffer sizing in large scale networks with large diameters (such as spatial networks) under comparable network-wide packet drop probabilities (QoS). However, using large buffer large deviations analysis, we show that network coding performs poorly against traditional queueing when it is not possible to have stochastic multiplexing with many other sources at intermediate nodes.

Finally, since network coding at the source may be used to dynamically buffer dropped packets at each fixed capacity link due to bursty fixed-rate arrivals at each source, we would like to also examine the dual scenario where the source rate (TCP window size) is controlled to deliver the maximum average throughput in a time-varying noisy wireless link (with varying information theoretic capacity) shared by many TCP connections. We show that network coding at the source promises an orderwise improvement in the mean TCP window size distribution as compared to the case where network coding is not used.

Table of Contents

List of Figures

Chapter 1

Introduction

Even until the very end of the 20th century, the task of analysis and design of telecommunication networks was cleanly divided between those who were focussed towards extracting the maximum capacity from each individual potentially noisy link connecting two terminals, and those who were concerned with managing the flow of randomly arriving data at hundreds and thousands of such terminals connected to each other in a network of links.

The latter study was founded on the discipline of stochastic analysis of networks, encompassing queueing theory, which makes no distinction between networks of cars, of units of production in a factory, or of packets of data (such as the Internet); whereas the former problem (that of reliably transmitting data over noisy links) was systematically explored by physicists and information theorists. From the perspective of network design and control, the capacity of a link is always a fixed quantity, bounded by underlying physical parameters which could be the width of a road (for cars), the speed of a machine (for factory production) or the capacity of a telecommunication link; that the link-designer provided the network designer.

Anantharam and Verdu [1] studied the information capacity of queues (and the timing information contained therein) in 1996. Subsequently, [2] and [3] examined the entropy of such queues. However, all of this work was focussed on understanding the information contained between batches processed by queues, rather than by the elements within the queues themselves.

The fundamental insight motivating the area of *network coding* is the observation that simple forwarding of packets by intermediate routers is inefficient and this inefficiency can be addressed by treating packets (which are vectors of binary symbols) as numbers rather than as jobs that need to be processed.

The intermediate routers in a network could then perform arithmetic (code) across these numbers (packets) and then forward the coded versions of the packets in place of the original packets. This technique, called network coding, can then be used to understand

network capacity (not merely link capacity) in information theoretic terms. As such, network coding presents a plethora of new research and application problems which are of interest to both information theorists and network theorists; this is borne out by the explosion of network coding related research over the past few years.

In their seminal paper, Ahlswede, Cai and Yeung [4] prove that for networks where the min-cut max-flow rate cannot be achieved by simple forwarding of packets, coding incoming packets at intermediate routers (network-coding) can help achieve the max-flow min-cut rate under multicast for such networks. Further, they show that such a strategy is optimal.

Parallel to this line of research, Luby, Mitzenmacher, Shokrollahi, Spielman and Stemann [38] provide randomized constructions of linear-time encodable and decodable erasure codes, called Tornado codes, that perform extremely close to capacity. Based on these Tornado codes, Byers, Luby, Mitzenmacher and Rege [39] present a fully scalable scheme to access erasure coded data from various sources over a lossy network: the authors refer to it as a digital fountain. While it is true that in these schemes all intermediate routers do not perform arithmetic, the principle of motivating digital fountain codes is that sources need to perform coding to combat erasures encountered over any link in the network; in this sense, digital fountain codes may be thought of as network codes as well.

1.1 Algebraic Network Coding: A brief survey

Koetter and Médard in [5] present a systematic algebraic formulation for linear network codes. Recently, Jaggi et. al. [6],[7] discovered a polynomial time algorithm for centralized linear network coding at intermediate nodes of a Directed Acyclic Graph. However, centralized control and coordination between intermediate routers is required for the purpose of designing a linear network code in this scenario.

1.1.1 Random Linear Coding

Ho. et. al. in [8, 9] introduce a randomized strategy for decentralized network coding; where the authors suggest the use of Random Linear Codes (RLCs) that asymptotically (in the field size) achieve the linear network code rate. Since the intermediate routers can code randomly independent of other routers in the network, RLCs offer the means for decentralized design of network codes and form the basis for practical network coding schemes [10].

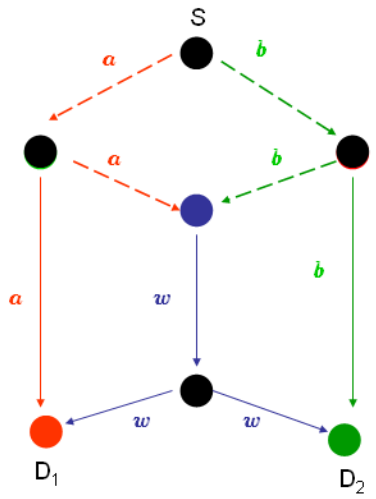

Figure 1.1: Random Linear Coding: the output of the intermediate node is $w = \alpha_1 a + \alpha_2 b$, where all symbols and operations are from the same finite field $\mathbb{F}_q$.

A simple introduction to RLC is provided by means of the following example. Consider the butterfly network shown in Figure 1.1, where source S wishes to multicast to D_1 and D_2. Since the min-cut of the network is 2, according to the result of Ahlswede, Cai, Li and Yeung [4], a multicast capacity of 2 should be achievable with linear network coding. Assume that the packets a and b are elements from some finite field $\mathbb{F}_q$ where q is sufficiently large. For instance, packets represented of length say k bits can be represented as elements of a the field $\mathbb{F}_{2^k}$.

The intermediate router codes the incoming symbols a and b by randomly choosing two elements $\alpha_1, \alpha_2 \in \mathbb{F}_q$ with uniform distribution over the field and performing the computation,

$$w = \alpha_1 a + \alpha_2 b$$

where addition and multiplication are defined according to the rules in $\mathbb{F}_q$.

D_1 receives two coded messages $Y_1 = a$ and $Y_2 = w$. We also assume that D_1 receives the coefficients α_1, α_2 either as a header or on a parallel channel. Ho et. al. [8] introduce schemes whereby this coefficient header decreases as a ratio of $\Theta(\log n/n)$ with packet-length n, and so we may neglect the overhead due to coefficient information for large packets. D_1 may now recover the original message symbols $a, b \in \mathbb{F}_q$ by solving the linear

equations $Y_1 = \hat{a}$ and $Y_2 = \alpha_1\hat{a} + \alpha_2\hat{b}$ for unknown variables $\hat{a}, \hat{b} \in \mathbb{F}_q$. The above system of linear equations is solvable if and only if the coefficient matrix

$$\begin{bmatrix} 1 & 0 \\ \alpha_1 & \alpha_2 \end{bmatrix}$$

is invertible. Since the elements of the coefficient matrix are randomly chosen, it can be shown that the coefficient matrix is invertible w.h.p. in q, the size of the field $\mathbb{F}_q$. Hence the achievable multicast network capacity approaches 2 asymptotically in q.

1.2 Network coding applied to networks

In the context of existing research on the capacity gains afforded by network coding, it becomes important to understand the benefits and limitations in the application of network coding to a variety of practical networks. In short – how good is network coding when applied to real networks? Classifying networks along the orders of topology (viz. Internet, sensor networks, peer-to-peer), data type (text, multimedia, real-time data) and transmission schemes (unicast, multicast, broadcast) and investigating the scope of network coding under these various scenarios will provide the fullest understanding to the above question. Our focus is on the impact of Random Linear Coding (RLC) on end-user performance (user impact) as well as network-wide resource allocation (network impact), in next-generation wireline and wireless networks. As a first step to understanding these problems, we examine scenarios where either (i) RLC is performed at all intermediate nodes (routers) or (ii) RLC is performed only at the source.

1.3 Network coding in intermediate nodes

The apparent advantage of network coding is that it makes the effective coded link rate at the output link of an intermediate router lower than the sum of the input link rates incident at that router. This suggests that in networks where links are priced as a function of packet rates across each link, network coding may offer substantial advantages in cost reduction.

The prospect of random mixing of data streams in place of the collision of data streams makes the application of RLC to multicast networks particularly attractive. Moreover, unlike the case of multiple unicasts between source-destination pairs in a network, where the problem of finding optimal network codes is known to be difficult [11], linear net-

work coding (and RLC) has been proven to be network min-cut capacity achieving (asymptotically capacity achieving, respectively).

1.3.1 Multicast cost minimization: wireline networks

Finding a cost minimizing flow allocation over the network using linear network coding turns out to be a convex optimization problem [12]. However, centralized solutions to a network-wide problem are unfeasible and hence much recent work has focussed on the problem of finding decentralized algorithms to determine the min-cost flow based on techniques such as primal-dual methods by message passing [13] between nodes in the network. However, this scheme requires a separate (differential-equation based) controller at each intermediate router for every flow passing through it. Each router requires a differential-equation based controller to evaluate the dual function, which itself is a complex problem.

Moreover, in most large-scale systems, end-users (or clusters of end-users) are likely to make routing decisions autonomously based on the price of the links [14],[15].

Thus the problem of min-cost routing becomes one of being able to first

- design a *pricing scheme at links* so that the network sum cost under user-equilibrium is the same as the global optimum, and thereafter
- design *simple greedy fully distributed algorithms* such that network-wide sum-cost is minimized even under autonomous user routing decisions.

The problem of min-cost multicasting in networks with selfish nodes is considered in Chapter 2. Based on our analysis, we also propose a simplified distributed algorithm for min-cost routing in multicast networks.

1.3.2 Unicast capacity maximization: wireless networks

The bulk of early research on network coding has been in the context of wireline networks. Due to the presence of wireless broadcast and interference, it is not possible to fully represent wireless network topology merely in the form of a connectivity graph. This makes analysis of the impact of network coding particularly difficult for wireless networks.

More recently, Gowaikar et al. [16] and Lun et al. [17] have studied some models of networks that abstract only the broadcast aspect of the wireless channel and show that

RLC based coding techniques can help achieve network capacity. However, this model does not incorporate wireless interference.

Since network coding can be used to achieve capacity in wireline networks, it is logical to wonder whether network coding may help achieve capacity in wireless networks.

- Do network codes help achieve capacity in wireless networks as well?
- If yes, then can these models be used to examine multicast capacity maximization as in Chapter 2?

Chapter 3 answers the first question in the affirmative for a mathematically abstracted wireless network, which we call a finite-field Broadcast and Additive Interference Network (BAIN); for such networks, we show that random linear coding achieves a unicast rate with a gap of $O(1/q)$ below the upper bound, where q is the size of the finite field.

Note that our result in Chapter 3 is only for the unicast case. The second question is still a subject of further investigation, till a multicast result can also be demonstrated over BAIN's.

1.4 Network coding at Source nodes

Optimal resource utilization is of great importance to network designers where the objective is to minimize the resource requirement to meet certain network design goals. The randomized spreading of information across the network makes RLCs attractive in cases where packet drops or losses are likely to occur in a network, such as in data dissemination and storage over large peer-to-peer networks [18],[19],[20]. For instance, under bandwidth constraints, the authors in [18] show that gossip via random linear coding reduces the delay of information dissemination. Analogously, for P2P file downloads under network capacity constraints, RLC between chunks of file-data is shown to improve robustness and delay of file downloads.

1.4.1 Spatial buffer multiplexing

The common underlying theme in much of the above work has been that network codes, and specifically RLC's, allow spatial (across the network) stochastic multiplexing across different flows and this feature can be utilized in improving reliability in large networks. A significant aspect of reliability in networks with time-varying loads is the presence

of buffers. It is known that buffer-sizes at nodes for large mesh networks need to scale orderwise with the size of the network in order to provide comparable packet-loss assurances.

Recently however, Lun, Medard and Effros [17],[21] exploit network codes for a capacity-approaching scheme for unicasts or multicasts over large networks. In their scheme, routers perform RLC over packets from different flows as well as over packets transmitted in previous time-slots. Further, for the case of Poisson traffic with i.i.d. losses at intermediate router queues (modelled as M/M/1 queues), they derive the packet error exponents in the large-delay regime. This is analogous to the use of block codes or convolutional codes for error control in the PHY that spread the information across multiple bits in a block or neighbourhood around each bit.

The insight from [17] that packets dropped in a particular time-slots can be recovered from RLCs containing the dropped packets in future time-slots motivates us to consider the following questions:

- Can we eliminate buffering at intermediate nodes in favour of coding only at the ends? We propose network coding at the source (correspondingly, decoding at the destination) with no buffers at intermediate nodes, thereby virtually sharing memory between links on a flow path. We call this *spatial buffer multiplexing* – where buffering and coding implemented at the source alone compensates for packet loss at any downstream bufferless link.
- Further, in the event of finite delays, how does network coding at the ends compare with queueing in intermediate routers? Here, we wish to compare QoS parameters such as delay and end-to-end packet loss probability (reliability) with coding as opposed to queueing.
- Ultimately, under comparable network-wide packet drop probabilities (QoS) how much of a network-wide memory resource gain does spatial buffer multiplexing provide over traditional buffering? How does this gain vary depending on the network topology?

We examine these questions by considering two complementary scenarios: first for the case of large networks with many flows through each node with finite buffer sizes at the sources (a many sources analysis), and second for the case of a network will a small number of flows with large buffers at the sources.

In Chapter 4 we use large deviations analysis to prove that for large-diameter networks (such as ad-hoc or sensor networks), large buffer gains are possible using spatial buffer multiplexing via RLC.

By contrast, for small diameter networks (e.g. most small-world networks such as the Internet), it is important to study the performance of RLC at the source even when it is not possible to have stochastic multiplexing with many other sources at a node. To study the effect on delay and packet loss in the case of a single bursty source-destination pair and compare queueing and coding in this regime, we study the large-buffer packet-drop probability (QoS) of coding and compare that against queueing. In the second half of Chapter 4, we specify conditions such that the same QoS requirements are met by the use of coding at the source instead of using a network buffer at the point of entry in the network. We show that network coding performs poorly compared to regular queueing in this scenario.

1.4.2 TCP NC

When TCP was designed, it was designed and optimized with the assumption that the networks that it was supposed to operate over have highly reliable node-to-node links so that dropped packets due to bad links are highly unlikely. It was this fact of the wired network that TCP utilized to build a congestion control mechanism; a dropped packet only meant one thing - a buffer overflow due to a congestion somewhere in the network. Thus, when the sender TCP algorithm is notified of lost packets, the additive increase and multiplicative decrease (AIMD) mechanism promptly cuts the transmission rate/TCP window size by half. Since wireless links have frequent errors, this causes the TCP window to go to 1.

In Chapter 4, we demonstrated how network coding at the source may be used to dynamically buffer dropped packets, at each fixed capacity link, due to randomly varying arrivals at each source.

This leads us to examine if network coding at the source can improve performance for the dual problem in Chapter 5, where the source rate (TCP window size) is controlled to deliver the maximum average throughput in a time-varying noisy wireless link.

For multiple TCP flows going through a wireless router, we show that we can obtain average TCP window size that is linear in the ergodic capacity of the wireless channel per flow.

Chapter 2

Minimum cost routing in Networks

2.1 Introduction

The single-source multicast problem for network coding has received much attention in recent years due to the tractability of designing optimal linear network codes for this case. Ahlswede, et. al. in [4] prove that for networks where the min-cut max-flow rate cannot be achieved by simple forwarding of packets, coding incoming packets at intermediate routers (network-coding) can help achieve the max-flow min-cut rate for such networks. Further, Ho et al. [8, 9] suggest the use of Random Linear Codes (RLCs) that can achieve the above linear network code rate asymptotically in the size of the symbol alphabet used for encoding/decoding. Since the intermediate routers can code randomly independent of other routers in the network, RLCs offer the means for decentralized design of network codes and form the basis for practical network coding schemes [10].

The problem of finding the minimum-cost multicast tree for networks has been studied extensively. For a general directed graph with a cost function at each edge, a specified root (source) and a subset of the nodes (receivers), the problem of finding a minimum-cost arborescence rooted at the source and spanning all the receivers is called the Directed Steiner Tree (DST) problem. Approximation algorithms for the DST, which is known to be NP-hard, has received considerable attention in recent years. Charikar et al. [22] present a non-trivial $O(i(i-1)k^{1/i})$ algorithm in $O(n^i k^2 i)$ time where k is the number of receivers. An LP-relaxation of the problem leads Zosin and Khuller [23] to a $O(D+1)$-approximation for the special case when the subgraph induced by the non-terminal nodes is a tree of depth D.

Lun et al. in [13] suggest a decentralized but cooperative scheme where the authors solve the network-coding min-cost optimization from [12] using primal-dual methods by message passing between intermediate routers. However, this scheme requires a separate (differential-equation based) controller at each intermediate router for every flow passing through it. Further, many current models of heterogeneous network service provisioning assume that selfish routing decisions are made by end-users based on the price of the links

[14],[15]. Such scenarios are likely to emerge with ad hoc or sensor networking where each end node is attached to a single multicast sink and therefore seeks to minimize its own cost. The dual problem of maximizing utility in a congestion game over a packet-forwarding network is considered in [14],[24]. Recently, the authors in [25] have framed this congestion control problem for network coding for single- and multiple-source multicasts as a generalization of the Eisenberg-Gale convex program to compute market equilibrium in the presence of economies of scale. Further, the primal-dual algorithm in [13] requires computationally intensive steps to be performed at each intermediate router.

In this chapter, we seek to design a min-cost flow-allocation algorithm when users are non-cooperative and minimize computation performed at each intermediate router. The users are assumed to be selfish agents that play a non-cooperative game to minimize personal costs selfishly without regard to the global or social optimal, and the expectation is that these users reach a Nash equilibrium if one exists. It is well-known that Nash equilibria do not optimize social welfare in general - a classical example of such an inefficient equilibrium is the 'Prisoner's Dilemma' [26]. Thus it immediately becomes important to quantify the inefficiency inherent in a selfish solution - dubbed the 'price of anarchy' [27, 28].

Dafermos and Sparrow [29] and Beckman [30] discuss the unicast selfish-agent min-cost routing problem in the context of transportation literature; this treatment corresponds to the uncoded packet forwarding scenario. Recently, [27, 31, 32] calculated the price of anarchy for this problem for a variety of convex cost functions for the capacitated and uncapacitated links. However, the optimization problem for the multicast min-cost flow with network coding as shown in the following section departs significantly from the min-cost unicast flow problem for uncoded packets and thus motivates independent analysis.

2.1.1 Main Contributions

In this chapter, we consider the min-cost flow routing problem with network coding for the selfish-agent case. We first consider the case with a single source and T multicast sinks (receivers), with each sink requiring a total rate R. We study the case where the network supports multi-path routing. A flow (along a particular path) from the source to a sink accumulates a cost that depends on the flow rate as well as the congestion on each of the links the flow traverses. Each sink t "steers" the flow rate allocation among its paths (i.e., among all paths from the source to the selected sink t such that the sum rate across paths is R) such that *its total cost* is minimized (in other words, a "greedy" setup for each

sink). We then generalize this framework to consider a multiple-sources scenario. The main contributions are as follows.

(i) We present the min-cost optimization problem for the single-source multicast with network coding and derive an asymptotically accurate approximation to that problem in Section 2.2. The selfish routing scenario is presented in Section 2.3 where a market is defined for bandwidth, being sold by links (sellers), that is utilized by flows to individual sinks (buyers).

We develop a mechanism for links (sellers) to allocate the link-costs among users of the link and demonstrate that for *monomial edge cost functions* (see section 2.3), a Nash equilibrium exists, and that the flow allocation at Nash equilibrium corresponds to the min-cost flow. Further, we show that capacitated links (i.e, links with capacity constraints) can be approximated arbitrarily closely using edge cost functions in the monomial class described in Section 2.3.

In other words, we show that the mechanism that we develop for link pricing leads to a rate allocation among users such that "greedy" flow rate allocation by each sink leads to a globally optimal flow rate allocation that minimizes the *total cost* in the network. In terms of algorithmic game-theoretic literature, this means that the 'price of anarchy' [27, 28] for the considered "greedy" system is 1.

(ii) In Section 2.4, we consider the multiple-source multicast problem and demonstrate a sub-optimal greedy scheme to achieve min-cost by selectively network coding within individual multicast sessions and not across sessions. (We note in passing that the general multi-source multi-destination network coding problem is intractable (NP hard) [11].)

(iii) Next, in Section 2.5 we present *UESSM*, User Equilibrium with Single Source Multicast, a non-cooperative decentralized flow-steering algorithm that provably converges arbitrarily close to a min-cost flow allocation for the class of convex, monomial edge cost functions defined in Section 2.3. At each receiver, UESSM "steers" flows across the paths leading to it in order to greedily minimize its own cost. This allows us to achieve the min-cost flow with network coding, without having to maintain state or perform per-flow primal-dual type calculations at every intermediate router. All that links have to do in UESSM is to allocate link costs according to the rule developed in subsection 2.3.1.

(iv) We next develop the Local Distributed Selfish Routing Algorithm (*LDSRA*) for min-cost routing. This algorithm is a local distributed algorithm where nodes in the network adjust flow fractions based on the local flow and cost information at each node and its neighbors. This is an analog of the Bellman-Ford algorithm, however, in the context of network coding. By using the end-to-end delay experienced by a probe packet as the marginal cost, LDSRA minimizes the total network latency (sum cost) by reallocating flows from the more expensive (greater delay) neighbor toward a cheaper (lower delay) one. We finally present simulation results for both UESSM and LDSRA to illustrate convergence properties.

2.2 Global Equilibrium

Consider a directed graph $G = (N, A)$ that models the network with the set of nodes N and the set of directed edges between them A. We consider a multicast session of rate R from source $s \in N$ to each of the sinks $t \in T, T \subseteq N$ implemented via multipath routing along the directed graph (network model). Flows along the set of paths $\mathcal{P}_t$ from s to t are indexed as $f_P \in \mathbb{R}$ for all $P \in \mathcal{P}_t$; $\mathcal{P} = \cup_{t \in T} \mathcal{P}_t$ is the set of all possible paths. Note that an edge may carry two or more flows to the same sink due to the presence of multipath routing.

We will associate a cost with the flow through each link on the network and formulate a global min-cost problem as one that minimizes the sum cost over the network. Accordingly, let $c_e(\cdot)$ be the cost function corresponding to edge $e \in A$ taking as argument a variable x dependent on the flows through edge e. We assume that the function $c_e(x)$ is strictly convex, positive, differentiable and monotonically increasing in variable x, with $c_e(0) = 0$. Further, we define the edge marginal cost $m_e(x) = c_e(x)/x$. We assume that the marginal cost is continuous and strictly increasing, with $m_e(0) = 0$.

Under traditional packet forwarding where packets are treated as objects, the cost of operating an edge in the graph is a function of the load of all packets that traverse that edge to all sinks. Most of the classical work on network optimization therefore deals with cost functions that take as argument the total fluid flow of packets to all sinks passing through that edge. That is, $x = \sum_{t \in T} \sum_{P \in \mathcal{P}_t} f_P$ and the corresponding edge cost incurred is $c_e(\sum_{t \in T} \sum_{P \in \mathcal{P}_t} f_P)$.

However, since we consider the case where intermediate routers perform random linear coding across packets to different sinks, it can be shown that the cost function $c_e(\cdot)$

takes as argument $x = \max_{t \in T} \sum_{P \in \mathcal{P}_t} f_P$ [12]. To see this intuitively, observe that under Random Linear Coding (RLC), packets to different sinks are linearly combined by the router to form a coded packet. In the fluid sense therefore, RLC allows for flows to different sinks to 'merge' to form the coded flow. This implies that on any edge, the effective size of the coded packet stream is dominated by the largest among net flows to each sink that traverse the edge.

Formally, the optimal cost for a rate R multicast connection from a single source $s \in N$ to sink nodes $T \subset N$ is given by the solution to the following optimization problem similar to [12, 13],

$$\begin{aligned}
&\text{GLOBAL}(G, c, R) \\
&\quad \text{minimize } C(f) &&= \sum_{e \in A} c_e(z_e) \\
&\quad \text{subject to } z_e &&= \max_t \left\{ \sum_{P \in \mathcal{P}_t : e \in P} f_P \right\} \quad \forall e \in A \\
&\quad f_P &&\geq 0 \quad \forall P \in \mathcal{P} \\
&\quad \sum_{P \in \mathcal{P}_t} f_P &&= R \quad \forall t \in T.
\end{aligned}$$

However, since $\max\{\ldots\}$ is not differentiable everywhere, motivated by the approach in [41],[13], we use the $\mathcal{L}_n$-approximation

$$\max\{x_1, x_2, \ldots x_k\} = \lim_{n \to \infty} \left(\sum_{i=1}^{k} x_i^n \right)^{1/n}$$

for analysis, thereby avoiding sub-gradient methods. Following the approximation of the max() above the $\mathcal{L}_n$-relaxed cost function of GLOBAL(G, c, R),

$$C_n(f) = \sum_{e \in A} c_e \left(\left[\sum_{t \in T} \Big(\sum_{P \in \mathcal{P}_t : e \in P} f_P \Big)^n \right]^{1/n} \right)$$

is differentiable everywhere. Formally, let C_n^* be the optimal solution to $\mathcal{L}_n$-GLOBAL(G, c, R) and C^* be the optimal solution to GLOBAL(G, c, R).

We note that this $\mathcal{L}_n$-approximation is motivated by the fact that as $n \to \infty$, $|C^* - C_n^*| \to 0$. Later (in Section 2.3) we will derive bounds on the approximation error for finite n for the class of functions considered in Section 2.3 (cf. Remark 1).

Since the cost functions are convex and the constraints form a convex set, the first-order Karush-Kuhn-Tucker conditions [53] are necessary and sufficient to solve $\mathcal{L}_n$-GLOBAL(G, c, R). We summarize the results in the following lemma.

Let $z_e^{(n)}$ be the corresponding $\mathcal{L}_n$ relaxation of z_e defined as

$$z_e^{(n)} \triangleq \left(\sum_{t \in T} x_{e,t}^n \right)^{1/n}$$

where

$$x_{e,j} \triangleq \sum_{P \in \mathcal{P}_j : e \in P} f_P$$

is the total flow of type j through the edge e.

Lemma 1. *A network coded multicast flow f^* is optimal for $\mathcal{L}_n$-GLOBAL(G, c, R) if and only if for all $t \in T$, and any paths $P_1, P_2 \in \mathcal{P}_t$ with strictly positive flows $f_{P_1}^*, f_{P_2}^* > 0$*

$$\sum_{e \in P_1} c_e'(z_e^{(n)*}) \alpha_{e,j}^{(n)*} = \sum_{e \in P_2} c_e'(z_e^{(n)*}) \alpha_{e,j}^{(n)*}, \tag{2.1}$$

for,

$$\alpha_{e,j}^{(n)} \triangleq \frac{z_e^{(n)}}{x_{e,j}} \cdot \frac{1}{\sum_{t \in T} (\frac{x_{e,t}}{x_{e,j}})^n}. \tag{2.2}$$

Proof: We append the cost function with the linear constraints via the Lagrangian multipliers λ_t, μ_P to form the Lagrangian

$$L(f, \lambda, \mu) = C_n(f) + \sum_{t \in T} \lambda_t \Big(\sum_{P \in \mathcal{P}_t} f_P - R \Big) - \sum_{P \in \mathcal{P}} \mu_P f_p.$$

We differentiate the Lagrangian with respect to each flow f_P, and the Lagrangian multipliers and equate each partial differential to zero to form a set of simultaneous equations in f, λ and μ. Solving these equations yields a minimizing solution f^*, λ^*, μ^*. Note that for all $P \in \mathcal{P}$, f_P^* and μ_P^* are complementary slack, i.e. $f_P^* \mu_P^* = 0$ with $\mu_P^* \geq 0$. Hence for paths with strictly positive flow, differentiating $L(f, \lambda, \mu)$ with respect to a particular flow f_{P_1}, for $P_1 \in \mathcal{P}_j$, gives

$$\sum_{e \in P_1} c_e'(z_e^{(n)}) \left(\frac{\sum_{P \in \mathcal{P}_j : e \in P} f_P}{z_e^{(n)}} \right)^{n-1} + \lambda_j = 0,$$

where $c'_e(x) = \frac{\partial c_e(x)}{\partial x}$. This implies that $\forall P_1, P_2 \in \mathcal{P}_j$ with $f^*_{P_1} > 0$,

$$\begin{aligned} &\sum_{e \in P_1} c'_e(z_e^{(n)*}) \left(\frac{\sum_{P \in \mathcal{P}_j : e \in P} f^*_P}{z_e^{(n)*}} \right)^{n-1} \\ &\leq \sum_{e \in P_2} c'_e(z_e^{(n)*}) \left(\frac{\sum_{P \in \mathcal{P}_j : e \in P} f^*_P}{z_e^{(n)*}} \right)^{n-1}. \end{aligned} \tag{2.3}$$

We are now done.

■

We note in passing that the behavior of

$$\alpha_{e,j} \triangleq \lim_{n \to \infty} \alpha_{e,j}^{(n)} \tag{2.4}$$

is not immediately clear – we cannot immediately state if the limit even exists. However, for any $n \in \mathbb{N}$ and any edge e with positive flows, we have from (2.2) that

$$\begin{aligned} \sum_{t \in T} \alpha_{e,t}^{(n)} &= \left[\frac{z_e^{(n-1)}}{z_e^{(n)}} \right]^{n-1} \\ &\geq 1 \end{aligned} \tag{2.5}$$

where the last inequality follows since the continuous function $L(p) = (\sum_i x_i^p)^{1/p}$ can be seen to be monotone decreasing in p for all $p \geq 1$ when all $x_i \geq 0$. Further, from Hölder's inequality,

$$\begin{aligned} \sum_{t \in T} x_{e,t}^{n-1}.1 &\leq (\sum_{t \in T} x_{e,t}^n)^{\frac{n-1}{n}} (\sum_{t \in T} 1)^{\frac{1}{n}} \\ &= (z_e^{(n)})^{n-1} |T|^{1/n}. \end{aligned}$$

Using the definition of $\alpha_{e,t}^{(n)}$ from (2.2) and the above inequality,

$$\begin{aligned} \sum_{t \in T} \alpha_{e,t}^{(n)} &= \frac{\sum_{t \in T} x_{e,t}^{n-1}}{(z_e^{(n)})^{n-1}} \\ &\leq |T|^{1/n} \end{aligned} \tag{2.6}$$

for every value of n. Then, from (2.6) and (2.5), it follows that

$$\lim_{n \to \infty} \sum_{t \in T} \alpha_{e,t}^{(n)} = 1.$$

2.3 Selfish routing and equilibrium

The solution to GLOBAL finds the optimum flow that minimizes routing cost in the overall network cost. This section deals with the system under the condition that each receiver minimizes its own cost to achieve user equilibrium under a defined bandwidth market to model selfish behavior as shown below. The ultimate goal of this section (and the next, respectively), is to show that under certain conditions on c_e, the user equilibrium corresponds to the global equilibrium (is comparable to the global equilibrium of a related optimization, respectively). These results will motivate a user-equilibrium based distributed optimization algorithm, discussed in Section 2.5.

2.3.1 The bandwidth market and link price-allocation

Each edge $e \in A$ sells bandwidth to the receivers (sinks) which are the users. Note that in the solution to the global problem we were merely concerned with the effective cost of the edge $c_e(z_e^{(n)})$ and did not need to consider how the cost of an edge in the network is divided among the flows through that network, while this sharing of costs (price allocation) needs to be defined for the user costs.

Hence, we propose a price allocation rule at each link and subsequently show that under this protocol, the sum cost under user equilibrium is equal to the min-cost for a wide range of cost functions c_e. Our price allocation rule is as follows. For each edge e the total cost of the flows $c_e(z_e^{(n)})$ is divided among flows of all type $t \in T$ so that $\frac{x_{e,j}^n}{\sum_{t\in T} x_{e,t}^n}$ fraction of the edge cost is borne by the flows in $f_P, P \in \mathcal{P}_j$ of type j. In turn $x_{e,j}$ is divided among all flows of type j through edge e in the ratio $f_P/x_{e,j}$ for all $P \in \mathcal{P}_j$. Thus the marginal cost of a flow f_P through a path $P \in \mathcal{P}_j$, $j \in T$

$$d_P^{(n)}(f) \triangleq \sum_{e\in P} c_e(z_e^{(n)}) \frac{1}{x_{e,j}} \frac{x_{e,j}^n}{\sum_{t\in T} x_{e,t}^n}. \tag{2.7}$$

Observe that by simply multiplying and dividing by $z_e^{(n)}$, (2.7) can be written as

$$d_P^{(n)}(f) = \sum_{e\in P} \frac{c_e(z_e^{(n)})}{z_e^{(n)}} \alpha_{e,j}^{(n)},$$

where $\alpha_{e,j}^{(n)}$ is as defined in (2.2).

2.3.2 User costs and equilibrium

Under the selfish condition, each flow from source s to destination j tries to minimize its marginal cost. This corresponds to each receiver minimizing its own total cost selfishly.

Since the cost functions are continuous and differentiable everywhere, we define *user equilibrium* as follows,

Definition 1. A *user equilibrium* is a flow allocation f feasible in $\mathfrak{L}_n$-GLOBAL(G, c, R) such that for any $P_1, P_2 \in \mathcal{P}_t$ where $f_{P_1} > 0$,

$$d_{P_1}^{(n)}(f) \leq d_{P_2}^{(n)}(f). \tag{2.8}$$

Note that this version of user equilibrium is also referred to as a *local Nash equilibrium* or *Wardrop equilibrium* in existing literature [29, 32].

Corresponding to this equilibrium, the total system cost for the flow f at Nash equilibrium is then

$$C_n(f) \triangleq \sum_{P \in \mathcal{P}} d_P^{(n)}(f) f_P.$$

In other words, any small $\epsilon \to 0$ change to the flow allocations from path P_1 to P_2 will only increase the sum cost along the paths in $\mathcal{P}_t$ for sink t. The notion of a local Nash equilibrium can be practically justified in scenarios where end users are in a distributed setting, with no or partial knowledge of the system, and try to reach their own local selfish optima by making small modifications to the flow allocations across paths in $\mathcal{P}_t$, where the flow steering proceeds only if that provides the selfish agent with immediate cost reduction.

2.3.3 User equilibrium vs. Global optimum

The similarity between the conditions in Lemma 1 and Definition 1 have been noticed for the case of costs depending on sum flows through an edge by Dafermos and Sparrow [29] and Beckman [30] and is cited by [27]. An important difference in our case is that while the edge cost in [27, 29, 30] is proportionally divided among all the flows through it, here, the cost is mainly borne by the sink with the maximum flow through the edge. The following lemma (adapted from [29, 30]) allows us to formulate the Nash equilibrium condition for a particular set of edge cost functions in terms of a global optimum for the same graph over *a different set of edge-cost functions*.

Lemma 2 ([29, 30]). *A single-source multicast flow f solves $\mathcal{L}_n$-GLOBAL(G, c, R) if and only if it is in local Nash equilibrium for $\mathcal{L}_n$-GLOBAL$(G, c'(x)x, R)$. Further, a local Nash equilibrium flow f exists for $\mathcal{L}_n$-GLOBAL(G, c, R). Moreover, if f and $\tilde{f}$ are feasible flows at Nash equilibrium, then $C_n(f) = C_n(\tilde{f})$.*

Proof: Comparing the KKT conditions from Lemma 1 with the user equilibrium conditions from Definition 1 leads us directly to the first statement of the Lemma. For the second statement, note that solving for a flow in local Nash equilibrium for $\mathcal{L}_n$-GLOBAL(G, c, R) corresponds to finding a (local) optimum flow for $\mathcal{L}_n$-GLOBAL(G, h, R), where $h_e(x) = \int_0^x c_e(t)/t \; dt$. Since c_e is continuous and monotonically increasing, h_e is strictly convex. Consequently, $\mathcal{L}_n$-GLOBAL(G, h, R) is a convex optimization over a convex set which implies that the optimum cost is unique, even though the solution points (the local minima) are not necessarily unique. ∎

This ensures that there exists a flow allocation that satisfies the user equilibrium (2.8).

We can now present the analog of the main result in Roughgarden and Tardos [27] for the min-cost multicast problem with network coding in the following theorem.

Theorem 1. *If for an instance $\mathcal{L}_n - (G, c, R)$ the cost function at each edge e is of the monomial form $c_e(z_e^{(n)}) = a_e(z_e^{(n)})^{k+1}$ for any fixed $k \in \mathbb{R},\ k > 0$, then for all $n \in \mathbb{N}$, the cost of flow f at local Nash equilibrium $C_n(f)$ equals the cost $C_n(f^*)$ of the global min-cost flow f^*.*

Proof: Since $c_e(z_e^{(n)})/z_e^{(n)} = a_e(z_e^{(n)})^k$ is monotonic increasing in $z_e^{(n)}$ for $k > 0$, we know from Lemma 2 that a Nash equilibrium exists. Further, note that for the given class of cost functions, the Nash condition (2.8) is the same as the KKT condition in (2.1). Hence, a Nash flow is also an optimum flow for the instance $\mathcal{L}_n - (G, c, R)$ and thus the cost functions are the same. ∎

We note that notwithstanding the simplicity of the proof, the above result is significant due to it's application in Section 2.5. The result above implies that for a large class of edge cost functions, a global min-cost multicast with network coding can be achieved by merely steering flows across edges to achieve user equilibrium corresponding to each sink t. In other words, the *price of anarchy* is 1.

Note that in general, the global min-cost flow can be achieved if each link charges the "Lagrangian cost" $h_e(x) = \int_0^x c_e(t)/t \; dt$ instead of the true cost $c_e(x)$. However, this would imply that the seller (link) earns an amount disproportionate to the true value of the goods or services (bandwidth) sold. The link-price allocation scheme detailed in subsection 2.3.1 ensures that the seller receives the 'fair' cost $c_e(x)$ but charges the selfish users differently so as to ensure that user equilibrium coincides with the socially-optimal flow allocation.

Observe that at $n = 1$, the $\mathfrak{L}_1$-GLOBAL(G, c, R) problem is the same as the classical min-cost flow-allocation problem. Also, correspondingly, our price allocation reduces to the allocation of link cost to a sink in linear proportion to the magnitude of flow to that sink through the particular link – thereby making the marginal cost of every flow through a link the same. This is exactly the same as the anarchic scenario in [27] where each flow through a particular edge e has the same marginal cost (edge delay) l_e and the net cost of that edge $c_e = l_e \sum_{e \in P, P \in \mathcal{P}} f_P$.

In general, the results herein define a differentiated pricing scheme for a shared service whose cost depends not on the sum of the demands but on the max demand. At the limit $n \to \infty$, we observe that only the set of users $T' = \arg\max_{t \in T} \sum_{P \in \mathcal{P}_t : e \in P} f_P$ pay for the cost of the link. Our price allocation rule automatically induces separate selfish agents to collaborate to benefit from this economy of scale.

Remark 1. *We now revisit the issue of the approximation error resulting from the $\mathfrak{L}_n$-relaxation of GLOBAL(G, c, R). Recall that in Section 2.2, the approximation was motivated by the fact that the error in the optimal cost (for any convex, increasing, differentiable link cost function) approaches 0 as $n \to \infty$. In this remark, we strengthen this statement by deriving bounds on the approximation error for finite values of n for the class of functions of the form $c_e(x) = a_e x^{k+1}$ considered in Theorem 1. For any given $\delta > 0$, we compute an $n(\delta)$ such that for any $n > n(\delta)$, the fractional approximation error (i.e., percentage error) $|C^* - C_n^*|/C^* \leq \delta$ is satisfied.*

Let f^* be a solution to GLOBAL(G, c, R) and f_n^* be a solution to $\mathfrak{L}_n$-GLOBAL(G, c, R). Further, let an optimal sum flow through edge e in the unrelaxed GLOBAL problem be denoted by $x_{e,t}^* \stackrel{\Delta}{=} \sum_{P \in \mathcal{P}_t : e \in P} f_P^*$.

Observe that for any vector $(x_{e,t})_{t \in T}$ of size $|T|$, if $z_e \stackrel{\Delta}{=} \max_{t \in T} x_{e,t}$ and $z_e^{(n)} \stackrel{\Delta}{=} (\sum_{t \in T} x_{e,t}^n)^{1/n}$, then we can bound the difference

$$z_e^{(n)} - z_e \leq (|T|^{1/n} - 1) z_e.$$

Since f^* is not necessarily an optimal flow allocation for $\mathcal{L}_n$-GLOBAL(G, c, R),

$$\begin{aligned} C_n^* &\leq \sum_{e \in A} a_e |T|^{(k+1)/n} (z_e^*)^{k+1} \\ &\leq |T|^{(k+1)/n} \sum_{e \in A} a_e (z_e^*)^{k+1} \\ &= |T|^{(k+1)/n} C^*. \end{aligned}$$

Thus, to ensure that $|C^* - C_n^*|/C^* \leq \delta$, we can solve for n to arrive at

$$n(\delta) > \frac{(k+1)\log|T|}{\log(1+\delta)}.$$

We note in passing that this bound is independent of the graph topology. We also refer the reader to Figures 2.1 and 2.2 in Section 2.5.5 which show that the $\mathcal{L}_n$-relaxation closely approximates the $\max(\cdot)$ *function for even small values of n.* ∎

2.3.4 Multicast over Capacitated Links

We next construct a feasible multicast over a set of capacitated links based on the above analysis. Let k_e be the capacity of edge $e \in A$. Suppose it is further desirable that most links in the network are loaded below $(1-\delta)$ factor of their capacities. This may be necessary to satisfy quality of service requirements, such as those on the average delay (note that in the presence of bursty traffic, the queueing delay becomes unbounded as the load approaches unity[1]).

To solve the above constrained multicast problem, we define the cost function of each edge, $e \in A$, as

$$c_e(x) \stackrel{\Delta}{=} \left(\frac{x}{k_e(1-\delta)} \right)^m.$$

[1]Recall that in the presence of stochastically time-varying flows (for instance, bursty packet rates from applications such as video/multimedia, flow connection initiations and terminations, etc), the average delay of a flow in a queueing system becomes large as the flow size gets close to the link capacity. This can be readily seen in an M/M/1 queue where the average delay increases as $1/\delta$. In general, for a GI/GI/1 queue, with inter-arrival times T of variance σ_T^2 and inter-service times X of variance σ_X^2, from Kingman's upper and lower bounds [54],

$$\frac{E[T]\sigma_X^2 - E[X](2-\rho)}{2(1-\rho)} \leq E[W] \leq \frac{\sigma_X^2 + \sigma_T^2}{2(1-\rho)}$$

we know that as the offered load ρ approaches 1 – that is as the mean arrival rate of data on a link is close to the link capacity – the mean waiting time for a packet entering the queue $E[W]$ scales in proportion to $(1-\rho)^{-1}$. The result can be extended from one queue to a Generalized Jackson Network of queues, where Gamarnik and Zeevi [76] demonstrate that in the heavy traffic limit as $\rho_n = 1 - k/\sqrt{n}$ for large n, the mean delay scales as $O(\sqrt{n})$.

Observe that as $m \to \infty$, edge costs will tend to zero for edges that satisfy the above constraints and become large for edges that do not. Hence, if there is a feasible flow allocation f for (G, c, R) over the constrained links, then the cost $C(f)$ at Nash equilibrium will be small, tending to zero as m becomes large. In fact, there exists m_0 such that for all $m > m_0$ the flow at Nash equilibrium satisfy the capacity constraints of all edges. More specifically, let $|A|$ denote the size of A (i.e., it is the total number of edges in the network), then an upper bound on m_0 is

$$m_0 = \frac{\log |A|}{-\log(1-\delta)}. \tag{2.9}$$

The reason for this is as follows. Since there exists a feasible flow f satisfying the above constraints, the global min-cost with the edge cost functions as above is at most $|A|$. Now, from Theorem 1, the cost of flow $\hat{f}$ at Nash equilibrium is the same as the global min-cost. Hence, it is too at most $|A|$. However, if $\hat{f}$ were to assign flow to an edge greater than its capacity, the cost on that edge alone will be at least $1/(1-\delta)^m$, which would be greater than $|A|$ for any $m > m_0$, where m_0 is as in (2.9).

2.4 Multiple Multicasts

In this section, we generalize the single-source multicast problem to the multiple-multicast sessions problem where each session corresponds to a source node taken from the set $S \in N$. Within each multicast session $s \in S$, network coding is performed across packets destined for sinks in set $T_s \subseteq N$. However, packets are not encoded across sessions to ensure computationally tractable decoding at each sink. So, each sink $t \in T_s$, can steer it's flows across the set of paths $\mathcal{P}_t^s$ from source s to sink t so as to deliver a total rate of R_t^s. As before, the total set of paths $\mathcal{P} \stackrel{\Delta}{=} \bigcup_{s \in S} \bigcup_{t \in T_s} \mathcal{P}_t^s$.

We can then formulate the min-cost problem M-GLOBAL(G, c, R) for multiple multicasts in the same way as [25]:

M-GLOBAL(G, c, R)

$$\begin{aligned} \text{minimize } C(f) &= \sum_{e\in A} c_e(z_e) \\ \text{subject to } z_e &= \sum_{s\in S} z_{e,s} \quad \forall e \in A \end{aligned}$$

$$z_{e,s} \triangleq \max_{t\in T_s} \left\{ \sum_{P\in\mathcal{P}_t^s : e\in P} f_P \right\} \quad , \quad f_P \geq 0 \quad \forall P \in \mathcal{P}$$

$$\sum_{P\in\mathcal{P}_t^s} f_P = R_t^s \quad \forall s \in S, \forall t \in T.$$

The corresponding $\mathcal{L}_n$-relaxed cost function for M-GLOBAL(G, c, R) is

$$C_n(f) = \sum_{e\in A} c_e \left(\sum_{s\in S} \left[\sum_{t\in T_s} (\sum_{P\in\mathcal{P}_t^s; e\in P} f_P)^n \right]^{1/n} \right).$$

Differentiating the equivalent Lagrangian

$$\mathcal{L}(f,\lambda,\mu) = C_n(f) + \sum_{s\in S}\sum_{t\in T_s} \lambda_t^{(s)} \left(\sum_{P\in\mathcal{P}_t^s} f_P - R_t^s \right) - \sum_{P\in\mathcal{P}} \mu_p f_p$$

with respect to a particular flow f_{P_1}, $P_1 \in \mathcal{P}_j^\sigma$ and applying the limit $n \to \infty$, we observe that at the minimizing flow f, for all $P_1, P_2 \in \mathcal{P}_j^\sigma$,

$$\begin{aligned} &\sum_{e\in P_1} c_e'(z_e^{(n)}) \left(\frac{\sum_{P\in\mathcal{P}_j^\sigma : e\in P} f_P}{z_{e,\sigma}^{(n)}} \right)^{n-1} \\ \leq \; &\sum_{e\in P_2} c_e'(z_e^{(n)}) \left(\frac{\sum_{P\in\mathcal{P}_j^\sigma : e\in P} f_P}{z_{e,\sigma}^{(n)}} \right)^{n-1}. \end{aligned} \tag{2.10}$$

where

$$z_{e,s}^{(n)} = \left(\sum_{t\in T} (\sum_{P\in\mathcal{P}_t^s : e\in P} f_P)^n \right)^{1/n} \tag{2.11}$$

and $z_e^{(n)} = \sum_{s\in S} z_{e,s}^{(n)}$.

Analogously, for each $s \in S$, we define

$$x_{e,j}^{(s)} \triangleq \sum_{e\in P : P\in\mathcal{P}_j^s} f_P.$$

The edge cost at each edge $c_e(z_e^{(n)})$ is divided among each $s \in S$ in proportion to the $z_{e,s}^{(n)}$ max flow from each s through e, i.e. $c_e(z_e)\frac{z_{e,s}^{(n)}}{z_e^{(n)}}$ is the fraction of the cost picked up by the set of flows towards sinks $t \in T_s$. Further, each $t \in T_s$ picks up $\frac{(x_{e,j}^{(s)})^n}{\sum_{t\in T}(x_{e,t}^{(s)})^n}$ fraction of $z_{e,s}^{(n)}$, which in turn is divided among all flows on paths $P \in \mathcal{P}_t^s$ in the ratio $f_P/x_{e,j}^{(s)}$.

Each sink $t \in T$ attempts to selfishly minimize the cost of transmission to itself. Since the costs across different sessions $\{s : t \in T_s\}$ are additive at each edge and cost functions at each edge are convex, minimizing total cost at sink t is the same as minimizing the cost for each session individually. Hence the criterion for local Nash equilibrium for multiple-session multicasts can be summed up in the following definition.

Definition 2. *A flow f, feasible for the instance (G, c, R) with multiple-multicast sessions S, is in local Nash equilibrium if for all $\sigma \in S$ and $j \in T_\sigma$, for any $P_1, P_2 \in \mathcal{P}_j^\sigma$,*

$$\begin{aligned} & \sum_{e\in P_1} \frac{c_e(z_e^{(n)})}{z_e^{(n)}} \left(\frac{\sum_{P\in\mathcal{P}_j^\sigma : e\in P} f_P}{z_{e,\sigma}^{(n)}}\right)^{n-1} \\ \leq & \sum_{e\in P_2} \frac{c_e(z_e^{(n)})}{z_e^{(n)}} \left(\frac{\sum_{P\in\mathcal{P}_j^\sigma : e\in P} f_P}{z_{e,\sigma}^{(n)}}\right)^{n-1}. \end{aligned} \tag{2.12}$$

We observe the similarity between (2.10) and (2.12) analogous to that between (2.1) and (2.8). The following corollary follows analogously from the reasoning in Section 2.3:

Corollary 1. *If for an instance $\mathcal{L}_n - (G, c, R)$, the cost function at each edge e is of the power law form $c_e(z_e^{(n)}) = a_e(z_e^{(n)})^{k+1}$ for any fixed $k \in \mathbb{R},\ k > 0$, then the cost of flow f at local Nash equilibrium $C_n(f)$ equals the cost $C_n(f^*)$ of the global min-cost flow f^*.*

2.5 Distributed Algorithms for Min-cost flow

Section 2.3 demonstrates that the sum-cost of the edges with any uniform power-law edge cost function under user equilibrium is the same as the min-cost. This result lends itself readily to the construction of a simple non-cooperative optimal min-cost flow routing algorithm for a single-source multicast with network coding. The following section deals with the single-source multicast for sake of simplicity. It is easy to show that due to the separable and additive nature of the costs for the multiple-source multicast, we can run the same algorithm independently over each session to reach the user equilibrium in this case too.

In this section, we develop two algorithms: User Equilibrium with Single Source Multicast (UESSM) and Local Distributed Selfish Routing Algorithm (LDSRA).

UESSM is a non-cooperative decentralized flow-steering algorithm that provably converges to the min-cost flow allocation for the class of convex, monomial edge cost functions defined in Section 2.3. At each receiver, UESSM "flow-steers" among the paths leading to it in order to greedily minimize its cost. This allows us to achieve the min-cost flow with network coding, without having to perform per-flow primal-dual type calculations at every intermediate router.

The Local Distributed Selfish Routing Algorithm (*LDSRA*) for min-cost routing is a local distributed algorithm where nodes in the network adjust flow fractions based on the local flow and cost information at each node. This is an analog of the Bellman-Ford algorithm, however, in the context of network coding. By using the end-to-end delay experienced by a probe packet as the marginal cost, LDSRA minimizes the total network latency (sum cost) by reallocating flows from the more expensive (greater delay) neighbor toward a cheaper (lower delay) one.

2.5.1 UESSM: User Equilibrium with Single Source Multicast

The implementation of this algorithm, UESSM, assumes *flow routing* between the source and destination, where the source router encodes downstream hop-by-hop routing information into the IP-header, as can be implemented in IPv6. The intermediate routers in the network between the source and sink do not need to maintain state-information locally. All that the intermediate routers need to do is route packets along the outgoing edges corresponding to the hop-by-hop information embedded in each packet and network code across packets of the same type at each instant of time using a random linear code.

Also, each downstream packet aggregates the cost that it has paid along each edge on a particular flow path. For efficiency, this information need not be carried by every downstream packet, but only by representative packets at each iteration of the algorithm.

Algorithm: UESSM

Initialization: In our implementation, we will choose a $\epsilon > 0$ small enough [cf. Section 2.5.3] such that R/ϵ is a positive integer, and require that all flow rates be at-least ϵ (a "keep-alive" rate). Also, the flow allocations in our implementation are elements from a lattice

$\mathcal{L} = \{0, \Delta, 2\Delta, \ldots, R\}$, for some fixed $\Delta > 0$ such that R/Δ and ϵ/Δ are positive integers. Thus, we can initialize at any arbitrary point on this lattice. For instance, for each sink $t \in T$ we can initialize at $f_{P'_t} = R - (Q_t - 1)\epsilon$ for some $P'_t \in \mathcal{P}_t$ and $f_P = \epsilon$ for all $P \in \mathcal{P} \backslash \{P'_t\}$.

Step: Now, one of the sinks $t \in T$ is chosen at random. Let us label the paths $P_1, P_2, \ldots, P_{Q_1}$, $P_{Q_1+1}, \ldots, P_{Q_1+Q_2}, \ldots P_{Q_1+\ldots Q_{T-1}+1}, \ldots P_{Q_1+\ldots Q_{T-1}+Q_T}$ where Q_t is the number of paths from the source to destination t. For some fixed $\xi > 0$, a receiver t picks a pair of paths $P_{\sum_{i=1}^{t-1} Q_i + l}$ and $P_{\sum_{i=1}^{t-1} Q_i + m}$, for any $l, m = 1, 2, \ldots, Q_t$ with $l \neq m$. Denoting

$$P_{t,l} = P_{\sum_{i=1}^{t-1} Q_i + l} \tag{2.13}$$

$$P_{t,m} = P_{\sum_{i=1}^{t-1} Q_i + m} \tag{2.14}$$

if

(a) $d_{P_{t,l}}(f) > d_{P_{t,m}}(f) + \xi$

(b) $f_{P_{t,l}} \geq \epsilon + \Delta$

(c) $f_{P_{t,m}} \leq R - \Delta$

then, $f_{P_{t,l}} \leftarrow f_{P_{t,l}} - \Delta$, $f_{P_{t,m}} \leftarrow f_{P_{t,m}} + \Delta$. Conversely, if (a), (b) and (c) hold with $P_{t,l}$ and $P_{t,m}$ interchanged, then $f_{P_{t,m}} \leftarrow f_{P_{t,m}} - \Delta$, $f_{P_{t,l}} \leftarrow f_{P_{t,l}} + \Delta$.

Termination: No user can make any flow switch if and only if $d_{P_{t,l}}(f) - d_{P_{t,m}}(f) \geq -\xi$, $\forall t, l, m$ which are feasible (i.e., (b) and (c) above are satisfied). In other words, at termination, for any receiver t and any pair of flows l, m $d_{P_{t,l}}(f) - d_{P_{t,m}}(f) < -\xi$ if and only if $f_{P_{t,m}} = \epsilon$.

Note that we would like to distinguish between the terms 'step' and 'iteration' as follows. By a 'step', we will mean the sequence of operations defined in the algorithm above. On the other hand by an 'iteration' we mean a 'step' that results in a flow reallocation. This distinction is made because due to random selection, at a step no flow reallocation may occur. In the rest of this chapter, we will only count 'iterations'.

2.5.2 Asynchronous implementation

The implementation of the algorithm above does *not* require synchronous timing between the clocks at the various sink nodes but only requires that the clocks have the same cycle frequency. We assume that the path-delay timescale along the network (for the update of the path costs etc.) is negligible compared to the time-steps in which the algorithm

proceeds. Each sink $j \in T$ picks a random delay that is exponentially distributed before adjusting it's flows in the manner outlined in Subsection 2.5.1. Since the exponential distribution is a continuous time-distribution, the collision probability is small. Further, since all flow steering is implemented at the source, the source can be designed to sequentially adjust flows of each sink. This ensures that only one sink adjusts flows at a time in the asynchronous algorithm, thereby retaining the same features as the synchronous implementation. Henceforth, we will denote each source adjustment (reallocation) as an iteration of the algorithm.

2.5.3 Convergence of UESSM to the min-cost flow

In this section, we restrict ourselves to edge cost functions of the form $c_e(x) = a_e x^{k+1}$, $k > 0$, as discussed in Section 2.3. From Theorem 1, it follows that a global optimum is the same as the cost at a Nash equilibrium.

Recall that in UESSM we are restricting each flow to have a rate of at least ϵ. Now, we choose ϵ as follows: Given any $\alpha > 0$, we will choose ϵ such that

(i) $\epsilon > 0$

(ii) $\epsilon < R/Q_t \quad \forall t = 1, 2, \ldots, |T|$

(iii) $R/\epsilon \in \mathbb{N}$

(iv) $|C_n^* - C_n^*(\epsilon)| < \alpha$,

where C_n^* is the optimal cost to $\mathfrak{L}_n$-GLOBAL(G, c, R), and $C_n^*(\epsilon)$ is the optimal cost to $\mathfrak{L}_n$-GLOBAL(G, c, R) under the additional constraint that $f_P \geq \epsilon, \forall P \in \mathcal{P}$. Observe that under this restricted simplex, the GLOBAL problem is still convex and differentiable. Further, since the cost functions are differentiable and finite, and the constraint sets are convex, given any $\alpha > 0$, there exists an ϵ such that the above conditions hold. Also, let $f^*(\epsilon)$ be an optimal solution to $\mathfrak{L}_n$-GLOBAL(G, c, R) with the ϵ-restricted convex constraint set.

Next, for any destination $t \in T$ we will formalize the notion of an infinitesimal reallocation of flows from path $P_{t,l}$ to path $P_{t,m}$ as defined in Equations (2.13),(2.14). Recall that due to monomial cost edge function and the $\mathfrak{L}_n$-approximation, the global cost function $C_n()$ is differentiable at all points. Accordingly, we can define $\nabla C_n(f) = (\frac{\partial C_n(f)}{\partial f_P})_{P \in \mathcal{P}}$ to be the $|\mathcal{P}|$-sized vector whose elements are $\frac{\partial C_n(f)}{\partial f_P}$.

Further, we define *direction vectors* $\mathcal{E}$ to be the collection of all vectors of the form $e_{t,l,m} = [0, \ldots, 0, -1, 0, \ldots, 0, 1, 0, \ldots, 0] \in \{-1, 0, 1\}^{|\mathcal{P}|}$ where the $P_{t,l}$-th element is -1 and the corresponding $P_{t,m}$-th element is 1. (Note that the vectors are not necessarily linearly independent – for example, $e_{t,l,m} = -e_{t,m,l}$). Accordingly, an infinitesimal shift of flow from $P_{t,l}$ to $P_{t,m}$ is given by the inner product $\nabla C_n(f)^T.e_{t,l,m}$.

In the following, we will utilize the property that the gradient function $\nabla C_n(f)$ is Lipschitz over the the space of feasible flow vectors f.

Lemma 3. *$\nabla C_n(f)$ is Lipschitz in the space of feasible flow vectors f with Lipschitz constant $L(\epsilon)$.*

Proof: We refer the reader to [42]. ∎

Recall that the constraint set (set of feasible flow rates) is described by a convex set where the flows corresponding to each receiver t is constrained to lie on a $|Q_t|$-dimensional (scaled) simplex (i.e., for each $t \in T$, $\sum_{l \in Q_t} f_{t,l} = R$, $f_{t,l} \geq 0$). For each f in the constraint set, we denote $\mathcal{E}(f) \subseteq \mathcal{E}$ to be the set of *feasible direction vectors*, where any $e_{t,l} \in \mathcal{E}(f)$ satisfies the following: a Δ shift of flow in the direction $e_{t,l}$ from f leads to f' which is a feasible flow vector.

Lemma 4. *Fix any $\alpha > 0$. Then, choose the following parameters for the UESSM algorithm:*

(i) Let $\xi = \frac{\alpha}{2(k+1)R|\mathcal{P}|^2}$, where k is the exponent in the edge cost function ($c_e(z) = a_e(z^{(n)})^{k+1}$),

(ii) Choose any $\Delta \leq \min\{\epsilon/10, \frac{\xi(k+1)}{2L(\epsilon)}\}$ such that ϵ/Δ is a positive integer, and $L(\epsilon)$ is given in Lemma 3.

With the above conditions, the following holds: Suppose that the algorithm UESSM terminates at iteration M. Then, $|C_n(f^{(M)}) - C_n^*(\epsilon)| < \alpha$, where $C_n(f^{(M)})$ is the cost with flow allocation $f^{(M)}$, and $C_n^*(\epsilon)$ is the optimal cost of the convex problem $\mathcal{L}_n$-GLOBAL(G, c, R) under the constraint that for all $P \in \mathcal{P}$, $f_P \geq \epsilon$.

Proof: Let $f^*(\epsilon)$ be an optimal flow corresponding to solution $C_n^*(\epsilon)$ and $f^{(M)}$ denote the flow in the M-th iteration (termination) of the algorithm.

Since $C_n()$ is convex, it follows from the gradient formula that

$$\begin{aligned} & C_n(f^{(M)}) \geq C_n(f^*(\epsilon)) \\ & \geq C_n(f^{(M)}) + \nabla C_n(f^{(M)})^T.(f^*(\epsilon) - f^{(M)}). \end{aligned} \tag{2.15}$$

Now, recall $\mathcal{E}(f^{(M)})$ is the set of feasible direction vectors corresponding to flow $f^{(M)}$.

Claim: There exists non-negative $\pi_{t,l,m}$ such that

$$f^*(\epsilon) \quad = \quad f^{(M)} + \sum_{e_{t,l,m} \in \mathcal{E}(f^{(M)})} \pi_{t,l,m} e_{t,l,m} \tag{2.16}$$

Proof: For each $t \in T$, we define the vector γ_t to be a $\{0,1\}$ vector of dimension $\mathcal{P}$ where $\gamma_{t,l} = 1$ for all $l = Q_{t-1}+1, \ldots, Q_t$ and 0 other-wise (i.e., γ_t corresponds to the flows destined for receiver t).

We now decompose $f^*(\epsilon) - f^{(M)} = \sum_{t \in T}(f_t^*(\epsilon) - f_t^{(M)})$ where $f_t^* = f^*(\epsilon) * \gamma_t$, (and similarly for $f_t^{(M)}$) where the $*$ operation corresponds to term-by-term multiplication (thus, $f_t^*(\epsilon)$ corresponds to the flows for receiver t).

Now, consider the (scaled and shifted)) simplex of feasible flows to receiver 1, i.e., $\mathcal{A}_1 = \{\sum_{l=1}^{Q_1} f_{1,l} = R \text{ and } f_{1,l} \geq \epsilon\}$. Let $v_i, i = 0, 1, 2, \ldots Q_1$ be the vertices of the (scaled and shifted) simplex $\mathcal{A}_1$ (the n dimensional scaled and shifted simplex has Q_1 vertices, with each vertex having one component equal $R - (Q_1 - 1)\epsilon$ and all other components being ϵ).

As the set is a convex simplex, we have for some $a_i \geq 0, \sum_i a_i = 1$,

$$f_1^*(\epsilon) \quad = \quad \sum_{i=0}^{Q_1} a_i v_i$$

Now, we consider two cases:

Case (i): $f_1^{(M)}$ is an interior point of $\mathcal{A}_1$.

In this case, all directions vectors $e_{1,l,m}$ at $f^{(M)}$ are feasible, and the existence of non-negative of $\pi_{1,l}$ is immediate. All direction vectors are feasible for the following reason: We have by the algorithm description (and the explicit construction of the values given in the Lemma statement) that $f^{(M)}$ lies on the Δ lattice and ϵ/Δ and R/Δ are positive integers. Thus, $f^{(M)}$ lies in the interior of $\mathcal{A}_1$ implies that each component of $f_1^{(M)}$ has flow rate value of at-least $\epsilon + \Delta$, in which case all direction vectors are feasible.

Case (ii): $f_1^{(M)}$ is a boundary point of $\mathcal{A}_1$.

Now, note that for a simplex of dimension $Q_1 - 1$, all boundary points can be described by *(a)* points that lie strictly within the interior of a simplex of dimension k for some $k = 1, \ldots Q_1 - 2$ or *(b)* the boundary point lies on a vertex of the simplex (i.e, $k = 0$).

For case (a), without loss of generality, let the boundary point be in the interior of a simplex of dimension k with vertices $v_0, v_1, \ldots v_k$. Then,

$$f_1^{(M)} = \sum_{i=0}^{k} b_i v_i$$

for non-negative b_i such that $\sum_i b_i = 1$. Thus, we have

$$f_1^*(\epsilon) - f_1^{(M)} = \sum_{i=0}^{Q_1} a_i v_i - \sum_{i=0}^{k} b_i v_i$$

$$= \sum_{i=0}^{k} a_i v_i + \sum_{i=k+1}^{Q_1} a_i (v_i - v_k) + v_k (\sum_{i=k+1}^{Q_1} a_i) - \sum_{i=0}^{k} b_i v_i$$

Now, let

$$\tilde{f}_1 = \sum_{i=0}^{k-1} a_i v_i + \left(\sum_{i=k}^{Q_1} a_i \right) v_k$$

Then, note that since $\sum_i a_i = 1$, $\tilde{f}_1$ lies in the k-dimensional simplex with vertices $\{v_i, i = 0, \ldots, k\}$. We now have

$$f_1^*(\epsilon) - f_1^{(M)} = (\tilde{f}_1 - f_1^{(M)}) + \sum_{i=k+1}^{Q_1} a_i (v_i - v_k)$$

where, by construction, $a_i \geq 0$ and $\tilde{f}_1$ and $f_1^{(M)}$ lie on the k-dimension simplex, with $f_1^{(M)}$ in the strict interior of this simplex. Thus, all vectors within the simplex are feasible (i.e., the direction vectors corresponding to both $(v_i - v_k)$ and $(v_k - v_i)$ for $i = 0, 1, \ldots, k$ are feasible as $f_1^{(M)}$ in the strict interior), we can choose feasible directions with non-negative weights to move from $f_1^{(M)}$ to $\tilde{f}_1$. In other-words, $(\tilde{f}_1 - f_1^{(M)})$ can be expressed as a non-negative weighted sum of feasible direction vectors.

For case (b) where we are terminating at the vertex (say v_0) of the $Q_1 - 1$ dimension simplex, the existence of non-negative π_{t_l} follows because $f_1^*(\epsilon)$ is in the (ϵ)-constrained set, and the feasible directions include all directions of the form $v_i - v_0, i = 1, 2, \ldots, Q_1$ which span the simplex set.

The proof is analogous for all other receivers. ∎

Thus, from (2.15) and (2.16), we have

$$\begin{aligned} & C_n(f^{(M)}) \geq C_n(f^*(\epsilon)) \\ & \geq C_n(f^{(M)}) + \nabla C_n(f^{(M)})^T . \sum_{e_{t,l,m} \in \mathcal{E}(f^{(M)})} \pi_{t,l,m} e_{t,l,m} \end{aligned}$$

where $\pi_{t,l,m}$ are non-negative and $e_{t,l,m}$ are feasible. From the termination condition of UESSM, we now have that along *all feasible* directions, $d_{P_{t,m}}(f) - d_{P_{t,l}}(f) \geq -\xi$. This is due to the following reason: Suppose $e_{t,l,m}$ is a feasible direction. This implies a Δ flow reallocation is allowed from flow $f_{P_{t,l}}$ to flow $f_{P_{t,m}}$ at iteration M. However, by the statement of the Lemma, iteration M is the termination step. Thus, UESSM decides not to re-allocate from flow $f_{P_{t,l}}$ to flow $f_{P_{m,l}}$. This can happen due to one of two possibilities: *(A)* $d_{P_{t,l}}(f^{(M)}) \leq d_{P_{t,m}}(f^{(M)})$ (i.e., $P_{t,l}$ is already a "cheaper" path than $P_{t,m}$, so UESSM does not further decrease the rate along flow $f_{P_{t,l}}$), in which case we have $d_{P_{t,m}}(f) - d_{P_{t,l}}(f) \geq 0 > -\xi$. The other possibility is *(B)* where $d_{P_{t,m}}(f^{(M)}) \leq d_{P_{t,l}}(f^{(M)}) < d_{P_{t,m}}(f^{(M)}) + \xi$ (i.e., the cost along path $P_{t,l}$ is only "slightly" more expensive than path $P_{t,m}$, and thus, UESSM decides not to switch). In case *(B)*, we have $0 \geq d_{P_{t,m}}(f) - d_{P_{t,l}}(f) \geq -\xi$.

Now, because the edge cost function is of the form $a_e(z^{(n)})^{k+1}, k > 0$, we have

$$\nabla C_n(f^{(M)})^T . e_{t,l,m} \quad = \quad (k+1) \left(d_{P_{t,m}}(f) - d_{P_{t,l}}(f) \right)$$

Thus, from the termination condition, we have

$$\nabla C_n(f^{(M)})^T . \sum_{e_{t,l,m} \in \mathcal{E}(f^{(M)})} \pi_{t,l,m} e_{t,l,m} \geq -(k+1) \sum_{e_{t,l,m} \in \mathcal{E}(f^{(M)})} \pi_{t,l,m} \xi$$

(In other words, at termination, along all feasible directions, the negative gradient is small).

Now, due to the fact that $\pi_{t,l,m}$ are bounded by R (as the space is bounded) and the fact that $|\mathcal{E}(f^{(M)})|$ is finite (because the number of paths are finite, an upper bound is $2|\mathcal{P}|^2$), we can choose ξ as in the Lemma statement to ensure that the difference in cost is no more than α. ∎

Lemma 5. *For a given $\alpha > 0$, let us fix $\xi = \frac{\alpha}{2(k+1)R|\mathcal{P}|^2}$. Choose any (strictly) positive $\Delta \leq \frac{\xi(k+1)}{2L}$ such that ϵ/Δ is a positive integer, and $L = L(\epsilon)$ is given in Lemma 3. Further, choose $\beta = \Delta\left((k+1)\xi - \Delta L\right)$.*

Suppose that at iteration M, there exists a user $t \in T$, $P_{t,l} = P_{\sum_{i=1}^{t-1} Q_i + l}$ for some $l = 1, 2, \ldots, Q_t$ and $P_{t,m} = P_{\sum_{i=1}^{t} Q_i + m}$, for some $m = 1, 2, \ldots, Q_t$ (i.e. $P_{t,l}, P_{t,m} \in \mathcal{P}_t$) such that $d_{P_{t,l}}(f) - d_{P_{t,m}}(f) < -\xi$ and $f_{P_{t,m}} \geq \epsilon + \Delta$ (i.e., a Δ flow switch is feasible).

Then, we have that a flow switch of Δ from flows $P_{t,m}$ to $P_{t,l}$ ensures that $C_n^{(M+1)} - C_n^{(M)} < -\beta < 0$.

Proof: To prove that a flow readjustment will cause a reduction in the overall cost function, we will borrow some results from the proof of convergence of constant step-size descent algorithms. Specifically, we will use the techniques in [53, Props 1.2.3, A.24] to demonstrate that if the gradient of the cost function is Lipschitz over the state space of flows, then if the difference of marginal costs between the paths are outside a ball of size ξ, the net cost reduction following a $\Delta \leq \frac{\xi(k+1)}{2L}$ readjustment of flows will be at least by $\beta > 0$, where $\beta = \Delta\left((k+1)\xi - \Delta L\right)$.

Note that by considering a flow reallocation from $P_{t,m}$ to $P_{t,l}$, the direction of descent $e_{t,m,l} = [0, 0, \ldots, 0, -1, 0 \ldots 0, 1, 0 \ldots 0]$, where elements -1 and 1 correspond to paths $P_{t,m}$ and $P_{t,l}$ in the $|\mathcal{P}|$ length vector $e_{t,m,l}$.

Now,

$$\begin{aligned} C_n(f) &= \sum_{e \in A} a_e (z_e)^{k+1} \\ &= \sum_{e \in P} a_e (z_e)^{k+1} + \sum_{e \in A \setminus P} a_e (z_e)^{k+1}. \end{aligned}$$

Thus, differentiating with respect to flow f_P where $P \in \mathcal{P}_t$

$$\begin{aligned} \frac{\partial C_n(f)}{\partial f_P} &= \frac{\partial}{\partial f_P} \sum_{e \in P} a_e (z_e)^{k+1} + 0 \\ &= \sum_{e \in P} a_e (k+1)(z_e)^k \left(\frac{x_{e,t}}{z_e}\right)^{n-1}. \end{aligned} \tag{2.17}$$

Also, observe that from (2.7) and (2.17), for the considered class of cost functions $c_e(z_e) = a_e z_e^{k+1}$,

$$\begin{aligned} \nabla C_n(f)^T . e_{t,m,l} &= \frac{\partial C_n}{\partial f_{P_{t,l}}} - \frac{\partial C_n}{\partial f_{P_{t,m}}} \\ &= (k+1)\left(d_{P_{t,l}}(f) - d_{P_{t,m}}(f)\right) \qquad (2.18) \\ &< -(k+1)\xi \qquad (2.19) \end{aligned}$$

where the last step follows from the lemma flow condition.

From the descent lemma [53, Prop A.24], we have that if L is the Lipschitz constant for $\nabla C_n(f)$ over the space of f, then

$$\begin{aligned} & C_n(f + \Delta e_{t,m,l}) - C_n(f) \\ & \quad \leq \Delta \nabla C_n(f)^T . e_{t,m,l} + \frac{1}{2}\Delta^2 L ||e_{t,m,l}||^2 \\ & \quad < \Delta \left(-(k+1)\xi + \frac{1}{2}\Delta L ||e_{t,m,l}||^2 \right) \\ & \quad \leq \Delta \left(-(k+1)\xi + \Delta L \right) \end{aligned}$$

Choosing $\Delta \leq \frac{\xi(k+1)}{2L}$, (such that ϵ/Δ is a positive integer), we have the desired result. ∎

We are now ready to state the main result of this section.

Theorem 2. *Choose the parameters α, ξ and Δ given in Lemma 4. Then, UESSM converges in a finite number of iterations (at iteration M), with the termination condition satisfying $|C_n(f^{(M)}) - C_n^*| < 2\alpha$, where $C_n(f^{(M)})$ is the cost with flow allocation $f^{(M)}$, and C_n^* is the optimal cost of the convex problem $\mathfrak{L}_n$-GLOBAL(G, c, R).*

Proof: For each flow allocation f that is not at the terminal condition, by the description of UESSM, there exists at least one user $t \in T$ and some pair of paths $P_{t,l}, P_{t,m} \in \mathcal{P}_t$ such that $d_{P_{t,l}}(f) - d_{P_{t,m}}(f) > -\xi$ for which a Δ flow reallocation is feasible.

Then, we have from Lemma 5 any feasible flow reallocation will reduce the sum cost by at least $\beta > 0$, i.e. $C_n(f^{(s+1)}) - C_n(f^{(s)}) \leq -\beta < 0$ at iteration s. Hence, at each iteration until the termination condition is reached, the cost function decreases by at least β. The initial cost $C_n(f^{(0)})$ of the iterations is positive bounded and the cost is non-negative. This implies that the algorithm will terminate in a finite number of iterations. Finally, from Lemma 4 the termination condition satisfies $|C_n(f^{(M)}) - C_n^*(\epsilon)| < \alpha$.

Further, note that we have chosen ϵ such that $|C_n^* - C_n^*(\epsilon)| < \alpha$ holds. Hence, by the triangle inequality,$|C_n(f^{(M)}) - C_n^*| < 2\alpha$. ∎

Although, UESSM converges and has provably good convergence properties, UESSM requires the source to maintain path information for all paths from the source to the destinations. This motivates the design of a local distributed algorithm where nodes adjust flow fractions based on the local flow and cost information at each node. We present such an algorithm in the following subsection.

2.5.4 Local Distributed Selfish Routing Algorithm (LDSRA) for Min-Cost Routing

Local routing algorithms for cost minimization have been studied in the past in the context of ad hoc network routing protocols, such as STARA [43, 44]. Such an algorithm can be implemented with an exponential-forgetting estimation as in STARA to estimate marginal costs from the source to each downstream node in the network and adjust fractional allocation of flows at each node so as to minimize the local marginal cost.

The algorithm proceeds in two phases. In the first phase, each node s identifies a set of neighbors N_s^k to reach destination k. Also, each node intermittently transmits probe packets along N_s^k, which accumulate marginal costs along the paths, and the feedback from these are used to estimate the marginal cost $D_{s,n}^k(t)$ from s to k along each particular neighbor $n \in N_s^k$ at time t. The second phase is the flow reallocation phase. Each node compares the estimated marginal costs of flows to a particular destination and then shifts flow allocation by a fraction Δ from the neighbor with higher marginal cost to one with lower marginal cost.

It can be shown that under steady state a flow allocation is at a user equilibrium (cf. Definition 1) *if and only if* all utilized paths from each node to each sink have equal marginal costs (see for instance Lemma 6.1 [44]). Since the above flow reallocation phase achieves the latter objective, under steady state it will also be at a user equilibrium. Further, if all edge costs are monomials, from Theorem 1 it follows that the above flow reallocation will have globally minimum cost in steady state.

In our simulations discussed next, we assume that the rate at which nodes reallocate flows is much slower than the rate at which probe packets are generated and cost estimates are gathered by each node. This allows us to assume, for purposes of simulation, that the estimates $D_{s,n}^k(t)$ at each node s are ideal. Future work will focus on designing stochastic-approximation based algorithms for joint estimation and rate allocation in this distributed framework, as well as analyzing its convergence and the rate of convergence under these conditions similar to the analysis in [46].

2.5.5 Simulation results

We simulate UESSM over the classic 7-node butterfly network in [4, 13] with the edge costs shown in Figure 2.1 for a rate 1 multicast session from source S_1 to destinations D_1 and D_2. The links are marked with the edge cost functions $c_e(x)$. In this example, $\mathcal{P}_1 = \{f_1, f_2, f_3\}$ and $\mathcal{P}_2 = \{F_1, F_2, F_3\}$.

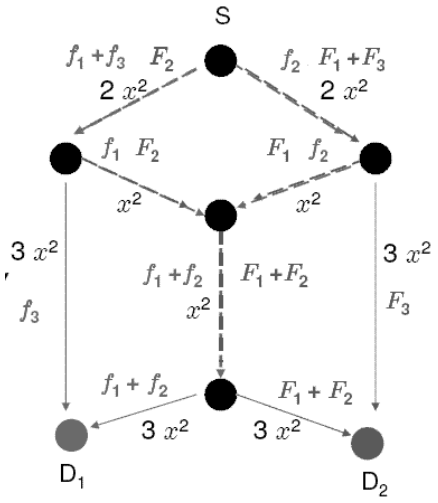

Figure 2.1: 7-node Butterfly network.

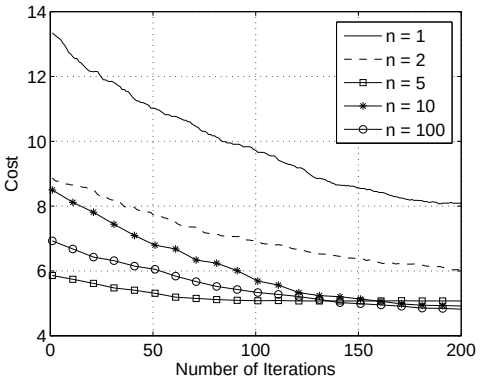

Figure 2.2: UESSM Algorithm: Comparing $\mathcal{L}_n$ approximation for $n = 1, 2, 5, 10, 100$.

We first study how $C_n(f)$ changes with increasing values of n in the $\mathcal{L}_n$-approximation to the max function. The trajectories for 100 representative UESSM runs with $\Delta = 0.01$ with varying values of n are plotted in Figure 2.2.

The $n = 1$ case corresponds to multicast without network coding and has a much higher sum-cost than that achieved by the $\mathcal{L}_{100}$-approximation, which is very close to the cost with using the non-differentiable max function in GLOBAL(G, c, R). However, we note that there is not much gain in going from $n = 10$ to $n = 100$. This suggests that the $\mathcal{L}_n$-approximation works well for even small values of n. Recall that we have bounded the minimum value of $n(\delta)$ given an approximation error target $\delta > 0$ in Remark 1.

We have also shown error bars corresponding to one standard deviation about the mean, with random initial conditions. We observe that, irrespective of initial conditions, the simulation sum-cost trajectories converges to the mean with progressively small variance. Typical trajectories of flow rates through various paths for the Butterfly network are presented in Figure 2.3 with a step-size of $\Delta = 0.01$.

We next provide simulation results with the LDSRA algorithm for the same Butterfly network. The costs under two $\mathcal{L}_n$-approximations (n = 1, 10) are plotted in Figure 2.4. We note that as expected, $\mathcal{L}_n$ cost decreases as n increases. Further, a comparison of Figures 2.2 and 2.4 verifies that both algorithms converge to the same sum-cost. Also, we

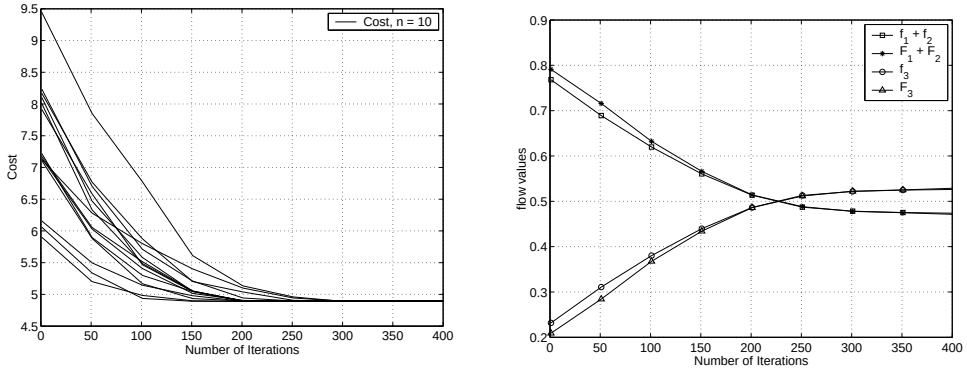

Figure 2.3: UESSM Algorithm trajectories: Sum costs and flows for the Butterfly network, $\mathfrak{L}_{10}$-GLOBAL(G, c, R), $\Delta = 0.01$.

compare the flows through the central edge in Figure 2.5 and observe that the equilibrium state corresponds to the symmetric min-sum cost flow allocation.

2.6 Conclusion

In this work, we have presented a cost splitting rule at each link for the min-cost problem using network coding and demonstrated that under this rule, the sum-cost across the network at user equilibrium is the same as the min-cost subject to the condition that all edges satisfy a uniform monomial cost function. Further, based on this result, we present two selfish min-cost routing algorithms - UESSM and LDSRA - which have desired performance in simulations. Additionally, we prove that UESSM converges to the min-cost flow allocation for any network topology.

Observe that in our discussion of multicast with many sources, we restricted the mixing of data to only between flows from a a particular sink. However, note that mixing between flows from different sources would involve designing a network code for the many-sources many-sinks problem. It is known that optimal code-design for such a case is NP-Hard [11]. Thus, the design and analysis of approximation algorithms for network coding with multiple-sources multicasting simultaneously would be an important area of future research.

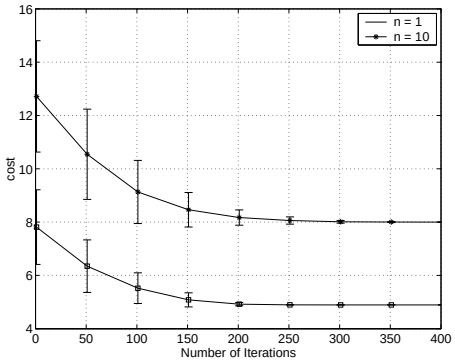

Figure 2.4: Butterfly Network with the LDSRA Algorithm: $\mathcal{L}_n$ approximation for $n = 1, 10$.

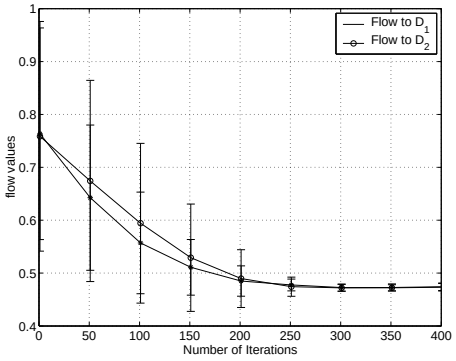

Figure 2.5: Butterfly Network with the LDSRA Algorithm: Flow allocation to central edge.

Chapter 3

Network Coding for Finite-Field Broadcast and Additive Interference Networks

3.1 Introduction

While network coding for broadcast networks has been the subject of much recent study [16], [33], [17], there remains a need to capture the interference nature of the wireless channel. In this work, we examine how network coding improves the throughput in a channel model that operates over a finite field but which incorporates both the interference and broadcast aspects of the wireless channel.

We consider a finite field interference network composed of nodes that are connected by links that are either wireline or wireless. Unlike packets (symbols) traversing the wireline links, symbols that are transmitted by a wireless node are subject to the broadcast constraint that all links carry the same symbol. Further, if two or more wireless nodes transmit symbols to a particular wireless receiver node, the symbols being sent over the air are subject to both channel fading and additive interference, where all channel and network operations are assumed to occur over an appropriate finite field. In the rest of this work, we will abuse the terms fading, and interference, to mean operations of multiplication by a random finite field element, and addition of symbols in the finite field, respectively.

Recently, Ray et al. [34],[35] have studied if source-channel separation exists in various networks in the presence of similar broadcast, and additive interference (albeit without fading). In networking literature, the classic multiple access interference model for collision multiple-access channel (MAC) has been Aloha [72, 73], where, a collision of two or more simultaneously transmitted messages results in a packet drop for both messages. Subsequently, the authors in [74], present a variation of Aloha which allows (simultaneous) multi-packet reception. In this work, we consider an intermediate stance where the interference of two signals in a MAC, both of which are elements of a particular finite field, is modeled as *the sum of the signals* in the same finite field. Symbol loss due to noise is modeled by allowing random complete erasure of the received signal.

Practical implications of such finite-field additive interference (for non-fading channels) are discussed in [35]. However the authors in [35] do not discuss the unicast capacity of a network composed of such finite field additive channels; the focus of their work is on the source-channel separation problem for a variety of broadcast and interference scenarios.

3.1.1 Main Contributions

(i) In Section 3.2 we introduce a finite field broadcast and additive interference network (directed graph) comprising of finite-field uniformly and independently distributed fading channels subject to broadcast and interference constraints, as well as random symbol erasure.

(ii) For a single-source unicast, we derive an achievable rate (4) using a random linear coding (RLC) strategy at each of the nodes, as well as an upper bound (Theorem 3) on the network capacity.

(iii) We show in Theorem 4 that our upper and lower bounds are tight asymptotically in the field size (i.e., the difference between the upper bound and the achievable rate scales as $O(1/q)$, where q is the field size).

(iv) We present an example network in Section 3.7 to demonstrate that the unicast capacity for broadcast and additive interference networks is lower in the absence of uniform i.i.d. fading. In other words, fading diversity can lead to large gains in unicast capacity for such network models.

We finally comment that the aim of employing the aforementioned finite-field model is to take a step towards determining the capacity region of wireless networks operating over a general Gaussian channel with fading. The latter, as is well known, is a very challenging problem – for even simple network configurations, such as the single-relay channel or the interference channel, the capacity regions are not yet known.

Hence, we consider a finite-field approximation of the general model, whose limit (under an appropriate distribution remapping) as the field size grows is the fading Gaussian channel. Even this simplification is not enough, as the capacity of a network of binary symmetric channels is not known, which is a special case of the finite-field approximation. Hence, we consider a further simplification of the model: instead of the additive noise term, we allow random complete erasure of the received signal. For this case, we are indeed able to obtain asymptotically tight bounds on the unicast capacity.

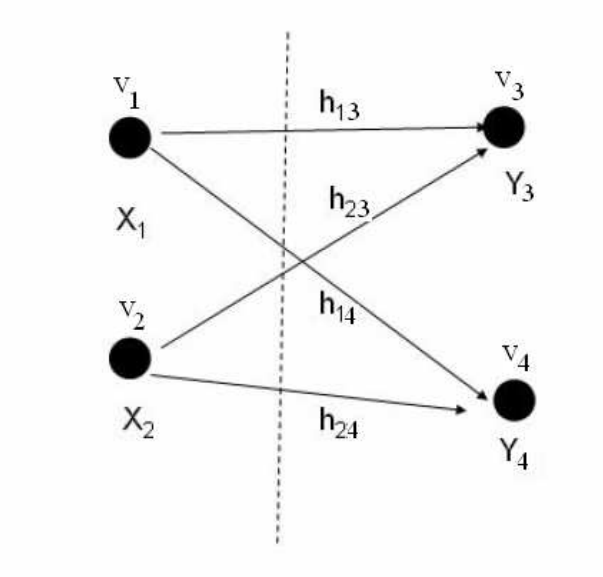

Figure 3.1: Model of a wireless channel with broadcast and interference constraints in the presence of fading coefficients $h_{ij} \in \mathbb{F}_q$. Node $v_i, i = 1, 2$, is constrained to send the same codeword (chosen from $\mathbb{F}_q$) on its outgoing links. Receiver $v_j, j = 3, 4$ decodes the symbol $Y_j = h_{1j}X_1 + h_{2j}X_2$ with probability $1 - \epsilon_j$ and erasure symbol $\mathcal{E}$ with probability ϵ_j.

This work is motivated by the expectation that the insights obtained from the simplified model will aid in the understanding of the more general model.

3.2 System Model and Notation: BAIN

We model a Broadcast and Additive Interference Network (BAIN) as a directed graph (DAG) $G = (V, E)$, where V, $|V| = N$, is the set of all nodes in the network, and for each $v_i, v_j \in V$ such that node v_i can transmit to v_j, there is a directed edge (link) $(v_i, v_j) \in E$. In this work, we restrict ourselves to directed acyclic graphs. Let $v_s \in V$ be the source node that wishes to transmit to destination $v_d \in N, v_d \neq v_s$. Further, let $\Gamma_O(v_i) \triangleq \{(v_i, v_j) | (v_i, v_j) \in A\}$ be the set of edges that leave node v_i. Correspondingly, $\Gamma_I(v_j) \triangleq \{(v_i, v_j) | (v_i, v_j) \in A\}$ is the set of edges incident on v_j. Hence, the out-degree and in-degree of any node v_j are $\delta_O(v_j) \triangleq |\Gamma_O(v_j)|$ and $\delta_I(v_j) \triangleq |\Gamma_I(v_j)|$, respectively.

Also, we model varying power constraints at various transmitters in the network by varying the entropy of the transmitted codewords at each transmitter. We consider slotted

time. Each $v_i \in V$ injects packets as a Bernoulli process with rate R_i. Then, we consider all codes to be subsets of the field $\mathbb{F}_q$ for any $\log q \geq \max_i\{R_i\}$. Thus, each codeword X_i transmitted by $v_i \in V$ must be an element of a subfield of $\mathbb{F}_q$, such that $H(X_i) = R_i$, $R_i \leq \log q$.

We assume that independent erasures occur at each receiver in the network. Note that in the BAIN, all erasures are node erasures, as opposed to edge erasures. The broadcast nature of the wireless channel is modeled by constraining all outbound edges $\Gamma_O(v_i)$ to carry the same symbol $X_i \in \mathbb{F}_q$.

We model interference as addition in the field $\mathbb{F}_q$ as follows: Consider the simplest multiple access network where two wireless nodes v_i, v_j transmit simultaneously to receiver v_r such that $\Gamma_I(v_r) = \{(v_i, v_r), (v_j, v_r)\}$. Let $X_i, X_j \in \mathbb{F}_q$ be the codewords transmitted by v_i and v_j, respectively. Let us consider coefficients h_{ij} uniformly distributed over $\mathbb{F}_q$. Then, v_r receives $Y_r \stackrel{\Delta}{=} h_{ir}X_i + h_{jr}X_j$, where all arithmetic is in $\mathbb{F}_q$, with probability $1-\epsilon_r$, and the erasure symbol $\mathcal{E}$ with probability ϵ_r. Erasure events are assumed to be independent across receivers.

Also, since G is a Directed Acyclic Graph (DAG), without loss of generality, we can arrange the nodes in topological order with $v_s = v_0$, $v_d = v_N$, and for each $(v_i, v_j) \in E$, $i < j$. Each node v_i – starting with v_0 – creates RLC's of the all the data packets that it possesses (i.e. the packets a node v_i has received over the edges in $\Gamma_I(v_i)$ for a node $i > 0$, or the original message data packets in case of the source node v_0) and sends them out over the edges $\Gamma_O(v_i)$ to the nodes in the next topological order. In the rest of this work, with some abuse in notation, we will write $v < u$ for any two $v, u \in V$ for a DAG $G = (V, E)$ when we mean that node u follows node v in topological order.

We next derive an upper bound and lower bound on the single-source unicast capacity of BAIN. Let v_s and v_d denote the source node and the destination node, respectively. Also, let C_q denote the unicast capacity of BAIN from v_s to v_d when operations and channel coefficients are in $\mathbb{F}_q$.

Remark 2. *Observe that the rate of information across any cut in the BAIN G is subject to the fading occurring at the edges that cross the cut. Due to the broadcast constraint imposed on the outgoing edges, it is also necessary for packets from each of the outgoing nodes to mix independently at the receiver nodes. For instance, if in a 2×2 cut of Figure 3.1, all h_{ij}'s are the same non-zero value, it can be seen that the rate across the cut is limited to $\log q$. However, if the vectors (h_{11}, h_{21}) and (h_{12}, h_{22}) are linearly independent, a rate of $2 \log q$ is achievable across the cut.*

3.3 Upper bound

We first derive an upper bound on C_q. To do so, we first define the following *graph transformation* from a BAIN to a Broadcast Erasure Network (BEN), which is the same as the Packet Erasure Network introduced in [16]. We will recall the BEN model in the following definition for sake of completeness.

Definition 3. *A Broadcast Erasure Network (BEN) is defined by a directed acyclic graph* $G' = (V', E')$ *where each edge* $(v'_i, v'_j) \in E'$ *represents a memoryless erasure channel and the constraint that all edges* $e' \in \Gamma_O(v'_i)$ *are constrained to carry the same symbol* X_i*. The random variable* $Z_{ij,t} = 1$ *indicates erasure on* (v'_i, v'_j) *at time* t*. The corresponding received symbol at time* t *on edge* (v'_i, v'_j) *is* $Y_{ij,t}$*.*

Note that in the above system, Y_{ij} is completely determined by the message symbols X_i and erasures Z_{ij}. Further, observe that there are no 'fading' coefficients associated with each channel in the BEN.

3.3.1 Transformation $\mathcal{T}$: BAIN $\rightarrow$ BEN

Given a BAIN $G = (V, E)$ we define the transformation $\mathcal{T} : (V, E) \rightarrow (V', E')$, where $\mathcal{T}(G)$ is a BEN, as follows.

(i) Initialize $V' = V$ and $E' = E$. Further, each $v_i \in V$ continues to transmit packets over its outgoing channels as a Bernoulli(R_i) process and subject to the broadcast constraint that the codewords on all outgoing channels from v_i must be the same at each time-slot.

(ii) For each node $v_r \in V$ that receives messages along edges in $\Gamma_I(v_r) \subset E$, create a node $v'_r \in V'$ and add an edge (v'_r, v_r) of rate $R_r = \log q$ to E', i.e. $V' := V' \cup \{v'_r\}$ $E' := E' \cup \{(v'_r, v_r)\}$.

(iii) For each edge $(v_i, v_r) \in \Gamma_I(v_r)$, $E' := E' \backslash \{(v_i, v_r)\} \cup \{(v_i, v'_r)\}$.

(iv) For any pair $(v_i, v_r) \in E \cap \Gamma_I(v_r)$ in the BAIN G, the erasure events in edges $(v_i, v'_r), (v'_r, v_r) \in E'$ in BEN $\mathcal{T}(G)$ are coupled, and are denoted by the same random variable $Z_{r,t}$. [1] Since the node erasures are Bernoulli, $Z_r(t) \sim$ Bernoulli(ϵ_r).

[1] so as to capture the node erasure process in the BAIN in terms of edge erasures. Note that while the main result in [70] is for i.i.d. erasure processes; an intermediate result, [70, Equation A.1] holds for correlated erasure processes as well. This is further elaborated in Remark 4 in Appendix A in [70].

Note that each receiver v'_r in $\mathcal{T}(G)$ receives separate signals over its incoming links, i.e., $\mathcal{T}(G)$ has broadcast constraints but no interference.

This capacitated broadcast erasure network $\mathcal{T}(G)$ will be referred to as the *equivalent broadcast erasure network* (EBEN) corresponding to G.

For a cut $(S, \bar{S})$ in $\mathcal{T}(G)$, we define the value $V_{\mathcal{T}(G)}(S)$ of the cut as

$$V_{\mathcal{T}(G)}(S) \stackrel{\Delta}{=} \sum_{\{i:v_i \in S, v_j \in \bar{S}, (v_i,v_j) \in E'\}} R_i \left(1 - \prod_{e \in (v_i, \Gamma_O(v_i))} \epsilon_e\right).$$

where ϵ_e is the edge erasure probability of edge $e \in E'$.

The capacity of broadcast erasure networks has been shown to be given by a *generalized* min-cut value in previous work by Gowaikar et. al. [16] and Lun et. al.[17]. We apply these results to the EBEN $\mathcal{T}(G)$ to derive an upper bound on C_q.

Lemma 6. *If the unicast capacity of the BAIN G is C_q and the unicast capacity of the corresponding EBEN $\mathcal{T}(G)$ is $C_q^{\mathcal{T}}$, then*

$$C_q \leq C_q^{\mathcal{T}} \tag{3.1}$$

Proof: To show that any rate acheivable by BAIN G is also achievable by EBEN $\mathcal{T}(G)$, consider any code $\mathcal{C}$ that achieves rate C_q in the BAIN.

Now, consider any fixed $v_r \in V$ in the BAIN.

Let each $v_i \in \Gamma_I(v_r)$ be transmitting codeword X_i over the channel $(v_i, v_r) \in E$ in the BAIN. Then, the corresponding codeword Y_r received by the v_r can be written as

$$Y_r = \sum_{v_i \in \Gamma_I(v_r)} h_{ir} X_i \mathbf{1}_{Z_r=0} + \mathcal{E}\mathbf{1}_{Z_r=1}$$

where $\mathcal{E}$ is the erasure symbol. We can now construct a code $\mathcal{C}'$ on the EBEN which is sample path coupled to code $\mathcal{C}$ on the BAIN as follows. Fix any sample path ω.

If at time t, $X_i(t,\omega)$ is the transmitted codeword by $v_i \in \Gamma_I(v_r)$, then the codeword transmitted at time t by $v_i \in \Gamma_I(v'_r)$ in the EBEN is $X_i(t,\omega)$. Observe here that by our construction of transformation $\mathcal{T}(\cdot)$, the set $\{v_i | (v_i, v_r) \in \Gamma_I(v'_r)\} \subseteq V'$ on the EBEN is the same as the set $\{v_i : (v_i, v_r) \in \Gamma_I(v_r)\} \subseteq V$ on the BAIN.

Thus, the codeword vector received by the intermediate node v'_r in the EBEN is $(X_i(t,\omega))_{i \in \Gamma_I(v_r)}$ if $Z_r(t,\omega) = 0$; and is the $|\Gamma_I(v_r)|$ length vector $(\mathcal{E}, \mathcal{E}, \ldots, \mathcal{E})$ otherwise.

In other words,

$$Y'_{ir}(t,\omega)\stackrel{\Delta}{=}X_i(t,\omega)\mathbf{1}_{Z_r(t,\omega)=0}+\mathcal{E}\mathbf{1}_{Z_r(t,\omega)=1} \tag{3.2}$$

is the codeword received on edge $(v_i, v'_r) \in E'$.

Let the coding $\mathcal{C}'$ at each intermediate node $v'_r \in V'$ be as follows: if Y'_{ir} is the codeword received on edge $(v_i, v'_r) \in E'$, then the codeword X'_r transmitted by v'_r is

$$X'_r(t,\omega) = \sum_{v_i\in\Gamma_I(v_r)} h_{ir}(t,\omega)Y'_{ir}(t,\omega). \tag{3.3}$$

if $Z_r(t,\omega) = 0$, and $X'_r(t,\omega) = 0$, otherwise.

Now, since the erasure process on edge $(v_i.v'_r)$ and (v'_r, v_r) are both coupled to the same random variable Z_r, the codeword Y'_r received by node v_r over edge (v'_r, v_r) in the EBEN $\mathcal{T}(G)$ is $\mathcal{E}$ if $Z_r = 1$ and X'_r otherwise.

Then, from equations (3.3) and (3.2),

$$Y'_r(t,\omega) = \sum_{v_i\in\Gamma_I(v_r)} h_{ir}(t,\omega)X_i(t,\omega)\mathbf{1}_{Z_r(t,\omega)=0}+\mathcal{E}\mathbf{1}_{Z_r(t,\omega)=1}$$

which is the same as $Y_r(t,\omega)$.Thus, the received codeword Y'_r at each node $v_r \in V'$ in the EBEN $\mathcal{T}(G)$ is sample path coupled to the received codeword Y_r at the corresponding node $v_r \in V$ in the BAIN G. This implies that rate C_q is achievable by code $\mathcal{C}'$ in the EBEN. We are now done. ∎

Theorem 3. *The unicast capacity, C_q, from source v_s to destination v_d in BAIN G consisting of links that are subject to broadcast and additive interference over the finite field $\mathbb{F}_q$, is upper bounded by $\bar{C}_q$, where*

$$\bar{C}_q = \min_{(S,\bar{S})\in\mathcal{S}(s,d)} V_{\mathcal{T}(G)}(S)$$

is the min-cut max-flow capacity of the BEN $\mathcal{T}(G)$, with $V_{\mathcal{T}(G)}(S)$ being the cut value for cut S.

Proof: From the previous lemma, it suffices to show that

$$C_q^{\mathcal{T}} \le \min_{(S,\bar{S})\in\mathcal{S}(s,d)} V_{\mathcal{T}(G)}(S). \tag{3.4}$$

Consider a block of n time-slots and any cut $(S, \bar{S}) \in \mathcal{S}(s, d)$. Let $S_{\text{edge}}\stackrel{\Delta}{=}\{v_i|\exists v_j \in \Gamma_O(v_i)\cap\bar{S}\}$ and $R_q^{\mathcal{T}}$ be any achievable information-theoretic unicast rate on the EBEN. We refer to an

intermediate result in [70, Equation (A.1)] in the derivation of the upper-bound for a BEN, reproduced here for sake of completeness: If $P_e^{(n)}$ is the average probability of block error for a block of length n,

$$nR_q^{\mathcal{T}} \leq \sum_{t=1}^{n} \sum_{v_i \in S_{\text{edge}}} I(X_i(t); (Y_j(t), j : (v_i, v_j) \in E', v_j \in \bar{S})) \\ +1 + nP_e^{(n)} R \tag{3.5}$$

where $X_i(t)$ is the codeword transmitted by $v_i \in V'$ and $Y_j(t)$ is the corresponding codeword received by v_j. Observe that due to the topological order imposed by the DAG $\mathcal{T}(G)$, the edges $(v_i, v_r'), (v_r', v_r) \in E$ cannot both be on the edge of a cut. Hence, for any $v_i \in S$, each outgoing edge $(v_i, v_j) \in E' : v_j \in \bar{S}$ must have independent erasures. Also, recall that the packet injection rate at v_i is R_i. Hence,

$$I(X_i(t); (Y_j(t), j : (v_i, v_j) \in E', v_i \in S, v_j \in \bar{S})) \\ = R_i(1 - \prod_{e \in (v_i, \Gamma_O(v_i))} \epsilon_e)$$

Then, from the definition of $V_{\mathcal{T}(G)}(S)$ and equation (3.5),

$$R_q^{\mathcal{T}} \leq V_{\mathcal{T}(G)}(S) + \frac{1}{n} + P_e^{(n)}$$

for any cut $(S, \bar{S})$ in $\mathcal{T}(G)$. Inequality (3.4) now follows directly if $P_e^{(n)} \to 0$.

We are thus done. ∎

We note that as a result of Lemma 6, the capacity of a similarly constructed BEN provides an upper bound for the non-fading case as well. (Here, set each $h_{ij} = 1$ at all times.)

Remark 3. *In the remainder of this work, we will create a wireline erasure network [cf. [17]] that supports unicast rate $C_q(1-O(1/q))$. Thereafter, we will construct a coupling between a set of packets over the BAIN and this wireline erasure network and show that that rate loss is limited to $O(1/q)$ to conclude that a unicast rate of $C_q(1-O(1/q))-O(1/q) = C_q(1-O(1/q))$ is achievable on the BAIN. This scheme is summarized in Figure 3.2.*

3.4 Min-Cut Max-Flow using RLC on a Tandem network

It is well known [16], [70] that for a wireline network with erasures, a max-flow min-cut rate is acheivable. The result was proved using random coding arguments and

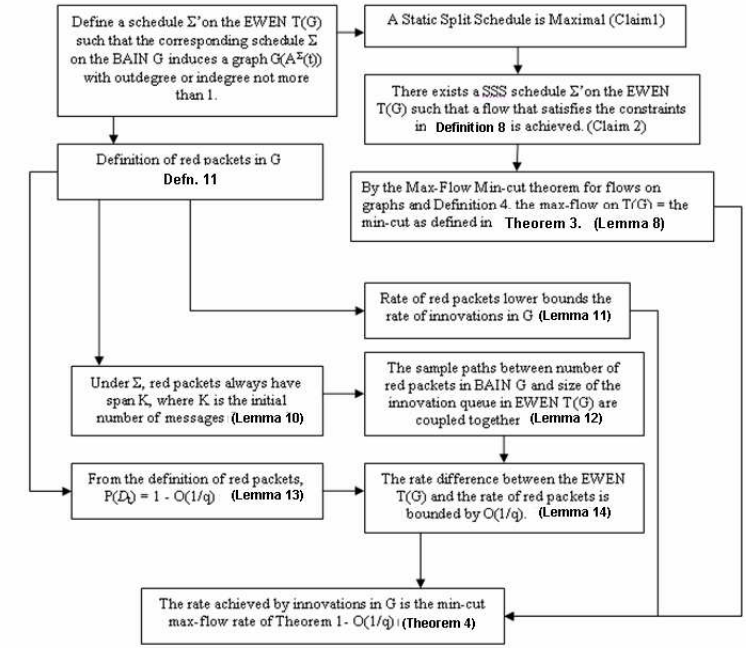

Figure 3.2: Summary of proof technique to obtain unicast achievable rate in BAIN.

subsequently [70], it was shown that a linear code is sufficient to obtain the cut capacity under the condition that the exact locations of the erasures were known at the destination.

In parallel, inspired by ideas from [37], Lun et. al. [17][71], provide fluid limit arguments to motivate that the max-flow min-cut rate can be achieved using random linear codes(RLC) where each node transmits the RLC of all packets currently present at the that node. As our system model is similar in construction to that of [71], we will first briefly summarize the min-cut max-flow acheivability results for a wireline erasure network (WEN) from [71]; and subsequently, in the following subsection, couple the innovation processes in the BAIN G with those of an appropriately defined equivalent WEN, $T(G)$. For sake of completeness, we present a formal and expanded version of the fluid-limit arguments presented in [71].

Lemma 7. *For a single path $L-1$-link tandem network $H=(N,A)$, $N=\{w_1,w_2,\dots w_L\}$ $A=(w_j,w_{j+1})|j=1,2,\dots,L-1$; with Bernoulli($r_j$) transmitter packet injection process at node w_j, and channel erasure process of Bernoulli(ϵ_{j+1}) on link (w_j,w_{j+1}) for $j=1,2,\dots,,L-1$, a rate $R_c\stackrel{\Delta}{=}(1-O(1/q))\times\min_j\{r_j(1-\epsilon_{j+1})\}$ is achievable using the coding strategy in Section 3.2.*

Proof: Let $\bar{m}\stackrel{\Delta}{=}(m_1,m_2,\dots m_K)$ be the message vector at source w_1 for any $K<(1-\delta)R_c\Delta(1-O(1/q))$ for any $\delta>0$ and $\Delta\in\mathbb{N}$. We will first show that in Δ time-slots, with probability probability approaching 1, as $\Delta\to\infty$ all K packets can be decoded at w_L.

Each w_j injects packets on (w_j,w_{j+1}) as an i.i.d. process $\{R_{jt}\}$ defined as

$$R_{jt}=\begin{cases}1 & \text{if a packet is injected on } (w_j,w_{j+1})\\ 0 & \text{otherwise}\end{cases}$$

with $\mathbb{P}(R_{jt}=1)=r_j$ at each time-slot t. Also by our model the error process $\{Z_{(j+1),t}\}$ on link (w_j,w_{j+1}) is an i.i.d. erasure process at each slot t with $\mathbb{P}(Z_{(j+1),t}=1)=\epsilon_{j+1}$.

Let us define sets $V_j(t)$ indexed for time-slots $t\in\mathbb{N}$ at each node w_j, for $j=2,\dots,L$. Let each $V_j(t)$ be the maximal linear independent subset of message packets at w_j at time t, in other words, $|V_j(t)|=|\text{span}(S_t(w_j))|$ where $S_t(w_j)=\{x_{tj}|j=1,2,\dots,t\}$ is the set of all packets received at w_j up to time t (without loss of generality, let us replace erasure symbols among the received packets by the 0 element from $\mathbb{F}_q$).

Definition 4. *For any $j=1,2,\dots,L-1$, a packet x transmitted from w_j to w_{j+1} is said to be* innovative *at w_{j+1} at time t if x is not an erasure symbol $\mathcal{E}$ and $x\notin span(S_t(w_{j+1}))$.*

The dynamics of $\{V_j(t)\}$ are as follows: if the incoming packet x is innovative, $|\text{span}(S_{t+1}(w_{j+1}))| = |\text{span}(S_t(w_{j+1}))| + 1$; correspondingly, $V_{j+1}(t+1) = V_{j+1}(t) \cup \{x\}$.

Since the $\text{span}(S_t(w_{j+1}))$ can only increase if $\text{span}(S_t(w_j)) > \text{span}(S_t(w_{j+1}))$, we can think of a queue build up of innovation $Q_j(t) \stackrel{\Delta}{=} |\text{span}(S_t(w_j))| - |\text{span}(S_t(w_{j+1}))|$. The first node where innovations are queued up is at node w_2, and the last node is at w_{L-1}.

Accordingly, a packet x received at w_{j+1} can be an innovative packet only if $Q_j(t) > 0$. Note however, that $Q_j(t) > 0$ does not guarantee that x is innovative. Since $x = RLC(S_t(w_j)) = \sum_{x_{tj} \in S_t(w_j)} \beta_{jt} x_{tj}$, it is possible that the random coefficient vector $\beta_{it} = 0$ corresponding to all packets $x_{ti} \notin \text{span}(S_t(w_{j+1}))$, thereby making the packet x non-innovative [i.e. the resulting $x \in \text{span}(S_t(w_{j+1}))$].

Definition 5. *The random coefficient vector $\{\beta_{jt}\}$ as defined above is said to be* unsuitable *if the following criteria are met:*

(i) If innovation is present in w_j with respect to w_{j+1}, i.e. $Q_j(t) > 0$ [2] and $RLC(S_t(w_j)) = x \in span(S_t(w_{j+1}))$.

(ii) Else if, innovation is not present in w_j with respect to w_{j+1}, i.e. $Q_j(t) = 0$, and $\Lambda_{jt} = 1$; where $\Lambda_{jt} \sim$ Bernoulli $(O(1/q))$.

Else, the coefficient vector is said to be suitable.

However, since the random coefficients β_{jt} are chosen independently of any other process in the network G the probability that the coefficient vector is unsuitable can be bounded by $1 - O(1/q)$(cf. [17], [18]). Thus, we can define an event A_{jt}, on probability space Ω, corresponding to the choice of coefficient vector $\{\beta_{jt}\}$ as follows $A_{jt} = 1\{\beta_{jt} \text{ is suitable}\}$.

Definition 6. *For any $j = 1, 2, \ldots, L-1$, a packet x transmitted from w_j to w_{j+1}, is defined as a* candidate *packet if it is received without erasure, and the random coefficient vector β_{jt} is not unsuitable.*

Remark 4. *Thus a candidate packet is also an innovative packet when $Q_j(t) > 0$ (i.e. the node w_j has innovation with respect to node w_{j+1}).*

[2] equivalently, $\text{span}(S_t(w_j)) \not\subseteq \text{span}(S_t(w_{j+1}))$

Noting from the above discussion, the packet injection process R_{jt}, the erasure process $Z_{(j+1),t}$ and the process A_{jt} at any time-slot t are independent of each other, and are each i.i.d. across time-slots.

Let $C_{jt} \in \{0,1\}$ be the random variable where $C_{jt} = 1$ if and only if a candidate packet is received on link (w_j, w_{j+1}). Then, by the definition of a candidate packet, $C_{jt} = R_{jt}(1 - Z_{(j+1)t})A_{jt}$, and thus the process $\{C_{jt}\}$ is i.i.d. across time with $\mathbb{P}(C_{jt} = 1) = r_j(1 - \epsilon_{j+1})(1 - O(1/q))$.

In the following, since w_2 is the first node where innovations are queued, we will first model the slotted-time arrival of innovative packets and candidate packets on link (w_1, w_2) as marked point processes in continuous time $\tau \in \mathbb{R}_+$ where arrivals occur at times $\tau \in \mathbb{N}$.

Packets arrive at w_2 at the beginning of a time-slot and, if serviced, depart at the end of that time-slot. We will assume that the source w_1 has an infinite backlog of innovative packets. Hence the innovation packet arrival process on (w_1, w_2) is the same as the candidate packet arrival process and is given by the i.i.d. geometric process $\{C_{1t}\}, t \in \mathbb{N}$. In terms of the continuous time model, this implies that the interarrival epoch τ_i between the $i-1$-th and i-th innovation is distributed as $\tau_i \sim \text{Geometric}(r_1(1 - \epsilon_2)(1 - O(1/q)))$.

In the slotted-time system, the arrival of a candidate packet at any w_{j+1} is governed by the i.i.d. process $\{C_{jt}\}, t \in \mathbb{N}$ defined above and depart at the end of the time-slot in which they are serviced. Therefore, in the continuous time model, the service-time distribution θ_{ji}^- for the i-th innovative packet at w_j is given by $\theta_{ji} \sim \text{Geometric}(r_j(1 - \epsilon_{j+1})(1 - O(1/q))$ where $\theta^- \triangleq \sup_\tau\{\tau < \theta\}$, i.e. the left limit point before the next epoch of length 1.

Accordingly, we can define the counting processes $B_j(\tau)$ and $C_j(\tau)$ for the arrival of innovative packets and candidate packets on edge (w_j, w_{j+1}) as follows:

$$B_1(\tau) = C_1(\tau) \triangleq \sup\{\nu | \sum_{i=1}^{\nu} \tau_i \leq \tau\} \tag{3.6}$$

$$C_j(\tau) \triangleq \sup\{\nu | \sum_{i=1}^{\nu} \theta_{ji} \leq \tau\} \tag{3.7}$$

$$B_j(\tau) \triangleq C_j(\int_0^\tau 1\{Q_j(s) > 0\}\mathrm{d}s) \tag{3.8}$$

for $j = 2, \ldots, L-1$, where the last relation follows from the observation in Remark 4.

By our definition of $Q_j(\tau)$ be immediately have that $Q_j = B_{j-1} - B_j$ for $j = 2, \ldots, L-1$. Let us define the processes $X_j \triangleq C_{j-1} - C_j$ and $Y_j \triangleq C_j - B_j$, where $C_1 = B_1$.

Note that the process Y_j corresponds to the total number of "idle" slots where the candidate packet could have been an innovative packet if only $Q_j(t) > 0$. We can therefore write the Skorohod problem for all $j = 2, 3, \ldots, L-1$ and all $\tau \geq 0$,

$$\begin{aligned} Q_j(\tau) &= X_j(\tau) - Y_{j-1}(\tau) + Y_j(\tau) && (3.9)\\ Q_j(\tau)dY_j(\tau) &= 0 && (3.10)\\ dY_j(\tau) &\geq 0 && (3.11)\\ Q_j(\tau) &\geq 0 && (3.12) \end{aligned}$$

with initial conditions

$$Y_j(0) = 0. \tag{3.13}$$

Consider the sequence of systems $(Q_j^N, C_j^N, X_j^N, Y_j^N)$ for $N \in \mathbb{N}$, where

$$\begin{aligned} Q_j^N(\tau) &\triangleq \frac{Q_j(N\tau)}{N}\\ C_j^N(\tau) &\triangleq \frac{C_j(N\tau)}{N}\\ X_j^N(\tau) &\triangleq \frac{X_j(N\tau)}{N}\\ Y_j^N(\tau) &\triangleq \frac{Y_j(N\tau)}{N} \end{aligned}$$

Observe that for the system of equations in (3.9) can be written in terms of the vector processes $\mathbf{Q} \triangleq [Q_j]_{j=2}^{L-1}$, $\mathbf{X} \triangleq [X_j]_{j=2}^{L-1}$, $\mathbf{Y} \triangleq [Y_j]_{j=2}^{L-1}$

$$\mathbf{Q}(\tau) = \mathbf{X}(\tau) + \mathcal{R}\mathbf{Y}(\tau)$$

where the reflection matrix $\mathcal{R}$ is given by

$$\mathcal{R} = \begin{bmatrix} 1 & 0 & 0 & 0 & \ldots & 0\\ -1 & 1 & 0 & 0 & \ldots & 0\\ 0 & -1 & 1 & 0 & \ldots & 0\\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\ 0 & 0 & \ldots & 0 & -1 & 1 \end{bmatrix}.$$

It is easy to check that $\mathcal{R}$ is an M-matrix [75, Section 7.2] since $G_R \triangleq (I - \mathcal{R})$ is a matrix will all positive elements and $\mathcal{R}$, being a lower triangular matrix is invertible.

Then, from the Oblique Reflection Mapping Theorem [75, Theorem 7.2], for each RCLL function $\mathbf{X}$, we can find the reflection mappings (Φ, Ξ) such that $\mathbf{Q} = \Phi(\mathbf{X})$, $\mathbf{Y} =$

$\Xi(\mathbf{X})$. In other words, for each $\mathbf{X}$, the pair $(\mathbf{Q}, \mathbf{Y})$ are unique. Further, Φ and Ξ are Lipschitz continuous on the space of RCLL functions endowed with the uniform norm.

Meanwhile, using the Strong Law of Large numbers for any finite $\tau \geq 0$, the following limits

$$\begin{aligned} c_j(\tau) &= \lim_{N\to\infty} C_j^N(\tau) \\ &= r_j(1-\epsilon_{j+1})(1-O(1/q))\tau \end{aligned} \tag{3.14}$$

exist almost surely and uniformly over compact sets for all $j = 1, \ldots L-1$.

Then by the Lipschitz property of the reflection mapping (Φ, Ξ),

$$\begin{aligned} Q_j^N(\tau) &\to_{\text{a.s.}} q_j(\tau) \\ X_j^N(\tau) &\to_{\text{a.s.}} x_j(\tau) \\ Y_j^N(\tau) &\to_{\text{a.s.}} y_j(\tau) \end{aligned}$$

almost surely, uniformly over compact sets, where the limiting functions satisfy the following set of "fluid" differential equations

$$\begin{aligned} q_j(\tau) &= (1-O(1/q))(r_{j-1}(1-\epsilon_j) - \\ &\quad r_j(1-\epsilon_{j+1}))\tau + y_j(\tau) - y_{j-1}(\tau) \\ q_j(\tau)dy_j(\tau) &= 0 \\ dy_j(\tau) &\geq 0 \\ q_j(\tau) &\geq 0. \end{aligned}$$

A formal proof of the above Functional Strong Law of Large Numbers for an open network is presented in [75, Theorem 7.23].

We can check that the following pair of functions (q, y) that satisfy the above set of equations

$$q_j(\tau) = (1-O(1/q))(r_{j-1}(1-\epsilon_j) - r_j(1-\epsilon_{j+1}))_+\tau \tag{3.15}$$

$$y_j(\tau) = (1-O(1/q))(r_{j-1}(1-\epsilon_j) - r_j(1-\epsilon_{j+1}))_-\tau; \tag{3.16}$$

for $j = 2, \ldots, L-1$ and are hence the unique solution to the system of fluid differential equations above.

After any fixed Δ timeslots, the total number of innovations that have reached the destination w_L is

$$\nu = B_1(\Delta) - \sum_{j=2}^{L-1} Q_j(\Delta).$$

Then for any $\delta_1 > 0$, there exists a Δ large enough such that

$$\begin{aligned} \frac{\nu}{\Delta} &\geq (1-\delta_1) \lim_{N\to\infty} \frac{B_1^N - \sum_{j=2}^{L-1} Q_j^N(\Delta)}{N\Delta} \\ &= (1-\delta_1)(1-O(1/q)) \min_{j=2,\ldots,L} \{r_{j-1}(1-\epsilon_j)\} \end{aligned}$$

The above expression implies that as long as $\delta_1 < \delta$, $K < \nu$ and thus the source messages can be decoded with probability $(1 - O(1/q))$. Thus, by choosing δ small enough, the resultant achievable decoded information-theoretic rate at the destination can be made arbitrarily close to R_c. ∎

3.5 Max-flow Min-cut on a WEN

Definition 7. *A wireline erasure network (WEN) $H = (V, E)$ is a directed graph with additional link properties (r_{jk}, ϵ_{jk}) corresponding to each arc $(v_j, v_k) \in E$; in each unit timeslot, v_j injects packets on (v_j, v_k) towards v_k with i.i.d. probability r_{jk} and the link (v_j, v_k) suffers packet erasures with i.i.d. probability ϵ_{jk} where the erasure process is independent of the packet injection process.*

Remark 5. *According to the definition above, the packet injection process, and the erasure process respectively, can alternately be stated as Bernoulli(r_{jk}), resp. Bernoulli(ϵ_{jk}).*

Let us consider the transform Θ on a wireline erasure network H, where each link (v_j'', v_j) in H, with capacity $\log q$ and erasure probability ϵ_j, is replaced by an erasure-free link of capacity $\log q(1-\epsilon_j)$, keeping the topology of both networks the same, viz. H.

As seen the previous subsection if H is a tandem wireline erasure network and a flow f is feasible on $\Theta(H)$, then an information-theoretic rate $f(1 - O(1/q))$ – in other words, an innovation flow of rate $f(1 - O(1/q))$ – is feasible on H.

In this subsection, we generalize the results from the previous subsection to the case any wireline erasure network with the topology of a directed acyclic graph. The authors in [71] present arguments to motivate the following result for any WEN H whose topology is directed acyclic graph. As before, for sake of clarity, we present a detailed proof based on the arguments in [71].

Lemma 8. *For any directed acyclic WEN H, if flow f is the maximum* unicast *flow feasible on error-free capacitated graph $\Theta(H)$, then an information theoretic rate $f(1-O(1/q))$ is achievable between the source and destination in the WEN H.*

Let $\mathcal{P}$ denote the set of all source-destination paths over the network $\Theta(H)$. Since the topology of H and $\Theta(H)$ are the same (only the link properties are changed) each $P \in \mathcal{P}$ is defined identically for both $\Theta(H)$ and H. Since the flow f is achievable on $\Theta(H)$, we can decompose the flow f as follows,

$$f = \sum_{P \in \mathcal{P}} f_P.$$

on each path such that $f_P \geq 0$ for all $P \in \mathcal{P}$.

Further, for any edge (v_j, v_k), we define the set of paths $\mathcal{P}_{jk} \triangleq \{P \in \mathcal{P} | (v_j, v_k) \in P\}$. Each path P in WEN H may now be viewed as a tandem network of length $|P|$. For any $v_j \in V$, let $\mathcal{P}_j \triangleq \bigcup_{v_k \in \Gamma_O(v_j)} \mathcal{P}_{jk}$ be the set of paths cross it.

Let $\bar{m} \triangleq \{m_1, m_2, \ldots, m_K\}$ be the set of messages at the source v_1. We will partition $\bar{m}$ into sets $\bar{m}_P$ of messages corresponding to each path P with $K_P \triangleq |\bar{m}_P| < (1-\delta) f_P (1 - O(1/q))\Delta$ for some $\delta > 0$ and $\Delta \in \mathbb{N}$.

We will now demonstrate that a rate of f_P innovations from source to destination is possible over the WEN H for all $P \in \mathcal{P}$. In other words, for any fixed $\delta > 0$, as $\Delta \to \infty$, all K_P innovations will be received at the destination with probability 1.

Let us first define the Bernoulli process $\{\psi_{jk}(t)\}$ corresponding to each edge $(v_j, v_k) \in V$ and each time-slot t, where $\psi_{jk}(t) \in \{P' | P' \in \mathcal{P}_{jk}\}$ with

$$\mathbb{P}(\psi_{jk}(t) = P) = f_P / \sum_{P' \in \mathcal{P}_{jk}} f_{P'}. \tag{3.17}$$

3.5.1 Coding scheme

Consider any node $v_j \in V$. For each $P \in \mathcal{P}_j$ traversing node v_j, the node maintains a separate set of packets $S_t^P(v_j)$ received on path P. Further, let us denote the next node on path P to be v_k, i.e. $(v_j, v_k) \in P$. At each time-slot t, the value of $\psi_{jk}(t)$ is known to both v_j and v_k. If $\psi_{jk}(t) = P$, node v_j creates as packet $x = RLC(S_t^P(v_j)) = \sum_{y \in S_t^P(v_j)} \beta_{jk}^P(t) y$ and forwards it to v_k on link (v_j, v_k).

Analogous to the treatment in the previous subsection, we will define sets $|V_j^P(t)| \triangleq \text{span}(S_t^P(v_j))$ at each node v_j and corresponding to each $P \in \mathcal{P}_j$.

Similarly, we can extend the notions of *candidate* and *innovative* packets from the previous subsection as follows. If x is received at v_k without erasure and the random coefficient vector $\beta_{jk}^P(t)$ is *suitable*, it is defined as a *candidate* packet on path P. Further, if $x \notin \text{span}(S_t^P(v_k))$ then x is said to be *innovative* at v_k on path P.

Accordingly, if v_k receives an innovative packet x in time-slot t on path P, $V_j^P(t+1) = V_j^P(t) \cup \{x\}$ and $\text{span}(S_{t+1}^P(v_k)) = \text{span}(S_t^P(v_k) + 1)$. Also, analogous to the case of the tandem network, a candidate packet x received at v_k over path P from node v_j can be innovative only if

$$Q_j^P(t) \stackrel{\Delta}{=} \text{span}(S_t^P(v_j)) - \text{span}(S_t^P(v_k)) > 0. \tag{3.18}$$

Note that the restriction of random linear coding only within packets received on the same path ensures that innovations do not mix between paths and hence the "innovation queue" processes Q_j^P can be decoupled at each node v_j.

3.5.2 Sample-path coupling

Let $R_{jk,t}, A_{jk,t}, Z_{jk,t} \in \{0,1\}$ be the indicator random variables representing the events that v_j injects a packet on link (v_j, v_k), the random coding vector $\beta_{jk}(t)$ is suitable (i.e. not unsuitable, c.f. Definition 5), and erasure occurs on link (v_j, v_k), respectively, all in time-slot t.

Also, for each $v_k \in \Gamma_O(v_1)$, let $\{B_1^P(t)\}$ be the i.i.d. arrival process of innovative packets on path P to node v_k such that $(v_1, v_k) \in P$. Further, recall that $\psi_{jk}(t)$ is the random variable taking values in $\{\mathcal{P}_{jk}\}$ where $\psi_{jk}(t) = P$ denotes the event that the link (v_j, v_k) is allocated for the purpose of bearing a packet from the set $S_t^P(v_j)$ on path P. Under the i.i.d. assumptions of the operations of packet injection, erasure, random linear coding, and path allocation on an edge; the overall sample space Ω (endowed with the appropriate probability measure $\mathbb{P}$) may be expressed as a product space as follows:

$$\Omega = \prod_{t \in \mathbb{N}} \prod_{(v_j, v_k) \in E} \Omega_{jk}(t)$$

where each $\Omega_{jk}(t)$ is the sample space induced by the random variables $\{R_{jk,t}, A_{jk,t}, Z_{jk,t}, \psi_{jk}(t)\}$ for $j > 1$; for each $v_k \in \Gamma_O(v_1)$, $\Omega_{1k,t}$ is the sample space induced by the random variables $\{R_{jk,t}, A_{jk,t}, Z_{jk,t}, \psi_{jk}(t), \{B_1^P(t) | P \in \mathcal{P}_{1k}\}\}$.

Corresponding to each $P \in \mathcal{P}$, we can then define the sample space Ω^P, endowed

with the same probability measure $\mathbb{P}$, given by the product space

$$\Omega^P = \prod_{t\in\mathbb{N}} \prod_{(v_j,v_k)\in P} \Omega^P_{jk}(t).$$

The sample spaces $\Omega_{jk}(t)$ and $\Omega^P_{jk}(t)$ are coupled as follows: for each $\omega^P \in \Omega^P_{jk}(t)$ $R_{jk,t}(\omega^P) = R_{jk,t}(\omega)$, $A_{jk,t}(\omega^P) = A_{jk,t}(\omega)$, $Z_{jk,t}(\omega^P) = Z_{jk,t}(\omega)$ and random variable $\psi^P_{jk}(t)(\omega^P) = 1$ if $\psi_{jk}(t) = P$.

Remark 6. *While the random processes $\{R_{jk}, A_{jk}, Z_{jk}, \psi^P_{jk}\}$, for a particular path P are all independent of each other, however, they are not independent across paths. In particular, for two paths $P, P' \in \mathcal{P}_{jk}$, the processes R_{jk}, A_{jk}, E_{jk} are the same, and if $\psi^P_{jk} = 1$, then $\psi^{P'}_{jk} = 0$ by construction of the spaces Ω^P and $\Omega^{P'}$.*

3.5.3 Path specific Skorohod Problems

The sample path map $\omega \to \omega^P$, by construction, preserves the i.i.d. packet arrival and service processes – formally, for link $(v_j, v_k) \in P$, $C^P_j(t, \omega^P) \stackrel{\Delta}{=} R_{jk,t}(1-Z_{jk,t})A_{jk,t}\psi^P_{jk}(t)$ and $C^P_j(t,\omega) \stackrel{\Delta}{=} R_{jk,t}(1 - Z_{jk,t})A_{jk,t}1\{\psi_{jk}(t) = P\}$ satisfy $C^P_j(t,\omega^P) = C^P_j(t,\omega)$; similarly $B^P_1(t,\omega^P) = B^P_1(t,\omega)$. Since the sample path evolution of the packet sets $\{S^P_t(v_j) : v_j \text{ on path } P\}$ is uniquely determined by the arrival and service processes, we can therefore state that $S^P_t(v_j,\omega) = S^P_t(v_j,\omega^P)$ and $Q^P_j(t,\omega) = Q^P_j(t,\omega^P)$.

Considering the dynamics of the tandem queue given by $\{Q^P_j | v_j \text{on path } P\}$ under the probability space $\{\Omega^P, \mathcal{F}^P, \mathbb{P}\}$, just as in the previous Section 3.4, we will now embed the discrete-time processes above in continuous time $\tau \in \mathbb{R}_+$ where packet arrivals at the node occur at times $\tau \in \mathbb{N}$ and packets are serviced at times $\tau = \theta^-$ where $\theta \in \mathbb{N}$.

Let $B^P_i(\tau)$ and $C^P_i(\tau)$ be the counting processes corresponding to the arrival of innovative packets and candidate packets, respectively, on path P, for any node v_i on path P which we will re-index as $i = \{1, 2_P, \ldots, j_P, \ldots, n\}$. We can analogously define processes $X^P_{j_P} \stackrel{\Delta}{=} C^P_{j_P-1} - C^P_{j_P}$ and $Y^P_{j_P} \stackrel{\Delta}{=} C^P_{j_P} - B^P_{j_P}$, where $C^P_1 = B^P_1$.

For each individual path P, we can then write the Skorohod problem analogous to the system in (3.9)-(3.13) on space Ω^P.

For each $P \in \mathcal{P}$, we can then consider an analogous sequence of systems indexed by $N \in \mathbb{N}$ and define the processes $(Q^{N,P}_{j_P}, C^{N,P}_{j_P}, X^{N,P}_{j_P}, Y^{N,P}_{j_P})$ on Ω^P. The corresponding fluid

limits can be shown to exist and satisfy analogous fluid differential equations where

$$\begin{aligned} c_{j_P}^P(\tau) &= \lim_{N\to\infty} C_{j_P}^{N,P} \\ &= \frac{f_P}{\sum_{P\in\mathcal{P}_{j_P,j_P+1}}} r_{j_P,j_P+1}(1-\epsilon_{j_P,j_P+1})(1-O(1/q))\tau. \end{aligned}$$

for each node (v_{j_P}) on path P.

Hence using Lemma 7, we can then show that for each path $P \in \mathcal{P}$, individually, a fluid rate of

$$\tilde{f}_P \triangleq \min_{(v_j,v_k)\in P} \left\{ \frac{f_P}{\sum_{P'\in\mathcal{P}_{jk}} f'_P} r_{jk}(1-\epsilon_{jk}) \right\} (1-O(1/q)) \tag{3.19}$$

can be supported.

Since this holds for every $P \in \mathcal{P}$ and the set $\mathcal{P}$ is finite, the joint convergence of the RCLL counting processes for all $P \in \mathcal{P}$ to the fluid processes holds; the corresponding fluid rate-set of $\{\tilde{f}_P | P \in \mathcal{P}\}$ is supported.

Therefore, on each edge (v'_j, v'_k) a net flow of

$$\tilde{f}_{j'k'} \triangleq \sum_{P\in\mathcal{P}_{j'k'}} \min_{(v_j,v_k)\in P} \{ \frac{f_P}{\sum_{P'\in\mathcal{P}_{jk}} f'_P} r_{jk}(1-\epsilon_{jk}) \} (1-O(1/q))$$

can be supported.

Since f can be any flow on $\Theta(H)$, in particular let f be the maximum flow possible for the network $\Theta(H)$. Then, there exists a cut $(S, \bar{S})$ (the min-cut corresponding to the max-flow) on $\Theta(H)$ such that for all $(v_j, v_k) \in E$,$v_j \in S, v_k \in \bar{S}$,

$$\sum_{P\in\mathcal{P}_{jk}} f_P = r_{jk}(1-\epsilon_{jk}).$$

This implies that the link (v_j, v_k) is the constraining link for every path $P \in \mathcal{P}_{jk}$; formally, for each $P \in \mathcal{P}_{jk}$ above,

$$(v_j, v_k) = \arg\min_{(v_l,v_m)} \{ \frac{f_P}{\sum_{P'\in\mathcal{P}_{lm}} f'_P} r_{lm}(1-\epsilon_{lm}) \}. \tag{3.20}$$

Hence, the corresponding flow of innovation supported across the cut $(S, \bar{S})$ is

$$\begin{aligned}
& \tilde{f}(S, \bar{S}) \\
= \quad & (1 - O(1/q)) \times \\
& \sum_{v_j \in S, v_k \in \bar{S}} \sum_{P \in \mathcal{P}_{jk}} \min_{(v_l, v_m)} \{ \frac{f_P}{\sum_{P' \in \mathcal{P}_{lm}} f'_P} r_{lm}(1 - \epsilon_{lm}) \} \\
\stackrel{(a)}{=} \quad & (1 - O(1/q)) \sum_{v_j \in S, v_k \in \bar{S}} \sum_{P \in \mathcal{P}_{jk}} \frac{f_P}{\sum_{P' \in \mathcal{P}_{jk}} f'_P} r_{jk}(1 - \epsilon_{jk}) \\
= \quad & (1 - O(1/q)) \sum_{v_j \in S, v_k \in \bar{S}} r_{jk}(1 - \epsilon_{jk}) \qquad (3.21)
\end{aligned}$$

where (a) follows from the property in equation (3.20).

Now since the RHS on (3.21) is the size of the min-cut $(S, \bar{S})$ on $\Theta(H)$, by the max-flow min-cut theorem on packet networks, $\tilde{f}(S, \bar{S}) = f(S, \bar{S})$(1 - O(1/q)).

We are now done. ∎

3.6 Achievable rate for BAIN

3.6.1 Coding scheme

We next lower bound the unicast capacity of BAIN with i.i.d. and uniform fading by constructing a coding strategy employing the same coding scheme as Lun et al. [17, 71]. Consider a coding epoch of Δ time units. Suppose that the source gets message packets at rate $\bar{C}_q(1 - \delta)$. Given a collection of messages $\{a_1, a_2, \ldots, a_m\}$, we define Random Linear Combining (RLC) of these messages by $RLC(\{a_1, a_2, \ldots, a_m\}) \stackrel{\Delta}{=} \sum_{i=1}^{m} \alpha_i a_i$ where each $\alpha_i, a_i \in \mathbb{F}_q$ and α_i's are chosen uniformly i.i.d. from $\mathbb{F}_q$. The source v_s now generates such RLCs, and transmits these RLCs to all receivers in $\Gamma_O(v_s)$. Similarly, each node v_i broadcasts RLCs of its received messages to its neighbours. The transmissions are synchronized to slotted time $t \in \mathbb{N}$; the exact schedule that the transmitters and receivers follow is detailed in Section 3.6.3.

Since each coded packet x is ultimately an RLC of the $a'_i s$, we can express $x = \sum_{i=1}^{m} \beta_i a_i$ where $\beta_i \in \mathbb{F}_q$. This vector $\beta = (\beta_i)_{i=1}^{m}$ is called the *auxiliary encoding vector* for packet x. We can now think of each node in the network to be forwarding *innovative packets* (i.e. new linear combinations of messages that were not in the span of the existing codewords at each receiver) and hence, as done in [17, 71], it suffices to track the flow of innovative packets through the network.

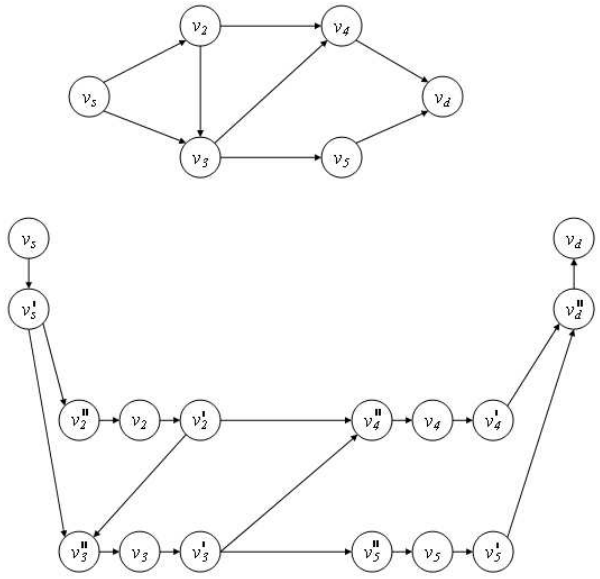

Figure 3.3: An example of a BAIN (above) and corresponding EWEN (below) obtained by transformation $T(\cdot)$.

3.6.2 Equivalent Wireline Erasure Network

We can now construct an equivalent wireline erasure network (EWEN) from the BAIN $G = (V, E)$ via the graph transformation T defined below.

Let V, V_T be sets of vertices and E, E_T be sets of directed edges, where $E \subseteq V \times V$ and $E_T \subseteq (V \cup V_T) \times (V \cup V_T)$. We can then define a graph $T(G)$ given by the transformation $T : V \times E \to (V \cup V_T) \times E_T$ on the BAIN $G = (V, E)$ as follows (see Figure 3.3):

i. Start with $V_T = \emptyset$, $E_T = \emptyset$;

ii. For each $u_i \in V$ such that $\Gamma_O(u_i) \neq \emptyset$, define vertex $u_i' \in V_T$, $E_T := E_T \cup \{(u_i, u_i')\}$;

iii. For each $v_j \in V$ such that $\Gamma_I(v_j) \neq \emptyset$, define vertex $v_j'' \in V_T$, $E_T := E_T \cup \{(v_j'', v_j)\}$;

iv. For each $(u_i, v_j) \in E$, $E_T := E_T \cup \{(u_i', v_j'')\}$, where u_i', v_j'' are as defined above.

Additionally, we will specify that the vertices are nodes (full duplex) and the edges are wired links with uniform capacity $\log q$. With a slight relaxation of rigour, will use $T(G)$ to denote both the network and the graph that describes the topology of the network – we

will call this the equivalent wireline erasure network (EWEN) $T(G)$. If for each node $u_i \in V$ that broadcasts symbols in $\mathbb{F}_q$ over BAIN G with rate R_i, we define a wired link (u_i, u_i') in network $T(G)$ where u_i transmits a packet with rate R_i.

Also, let Z_j be a random process (indexed by time t) as follows,

$$Z_j \triangleq \begin{cases} 0 & \text{if packet is received correctly} \\ 1 & \text{if decoder outputs erasure symbol} \end{cases}$$

at each $v_j \in V$, on BAIN G, with erasure probability $\mathbb{P}(E_j = 0) = \epsilon_j$. We will specify that the corresponding link $(v_j'', v_j) \in E_T$ is an erasure channel with a Bernoulli erasure process $\tilde{Z}_j$.

Remark 7. *Note that the transformation $T(\cdot)$ to construct the EWEN is different from the transformation $\mathfrak{T}(\cdot)$ to construct the EBEN. In the EBEN, the transmitting nodes were under the broadcast constraint (viz. each edge in $\Gamma_O(v_i)$ in $\mathfrak{T}(G)$ must carry the same message symbol at any timeslot). However, the EWEN is fully wireline, in the sense that each edge in $\Gamma_O(v_i)$ in $\mathfrak{T}(G)$ can carry different symbols at the same timeslot.*

3.6.3 A schedule on the EWEN and the BAIN

Let us introduce the notation $Q_i(t)$ for the sum of all individual path queues at node $v_i \in V$ in EWEN $T(G)$ defined as

$$Q_i(t) = \sum_{P \in \mathcal{P}_i} Q_i^P(t).$$

where $Q_i^P(t)$ is as defined in (3.18).

For each $v_i \in V$, we will now couple the progression of innovations $m(v_i, t)$ over G with the progression of innovative packets and the corresponding innovation queues $Q_i(t)$ in $T(G)$. Note that in the above statement, we restrict ourselves only to the nodes $v_i \in V$ (as opposed to nodes $v_k', v_k'' \in V_T$ which are not defined in BAIN G, but exist only in the EWEN $T(G)$).

Let us denote the unique directed path between two nodes $v_i, v_j \in V$ in $T(G)$ by $P(v_i, v_j) = v_i \to v_i' \to v_j'' \to v_j$.

Definition 8. *Let $\tilde{f}$ be a flow of innovation on the EWEN $T(G)$ that satisfies*

$$\tilde{f}_i^T \triangleq \sum_{P: P(v_i, v) \in P, v \in V} \tilde{f}_P < R_i(1 - \prod_{v_j \in \Gamma_O(v_i)} \epsilon_j) \tag{3.22}$$

and

$$\tilde{f}_j^R \stackrel{\Delta}{=} \sum_{P:P(v,v_j)\in P, v\in V} \tilde{f}_P < \log q(1-\epsilon_j). \tag{3.23}$$

for all $v_i, v_j \in V$.

Definition 9. *We define a schedule as a time-indexed collection of active path sets* $\Sigma'(\omega,t)$ *on EWEN* $T(G)$ *that satisfies,*

$$A^{\Sigma'}(\omega,t) \stackrel{\Delta}{=} \{P(v_i,v_j)|Z_j(\omega,t)=0\} \tag{3.24}$$

such that if $P(v_i,v_j) \in A^{\Sigma'}(t)$*, then* $P(v_k,v_j), P(v_i,v_k) \notin A^{\Sigma'}(t)$ *for all* $v_k \neq v_i, v_j$ *respectively in EWEN* $T(G)$.

Remark 8. *The schedule* Σ' *induces a subgraph* $T(G)(A^{\Sigma'}(\omega,t))$ *on the EWEN* $T(G)$ *such that the in-degree or out-degree of any node in* $T(G)$ *is no more than 1 at any time* t*. In other words, the nodes in* V_T *do not queue any packets.*

Since each node is full duplex, the transmitter $Tx(v)$ and the receiver $Rx(v)$ of any node $v \in V$ do not conflict with each other. Hence, we can define the bipartite graph $K(G) \stackrel{\Delta}{=} \{Tx(V), Rx(V), Tx(V) \times Rx(V)\}$, where edge $(v_i,v_j) \in Tx(V) \times Rx(V)$ if there exists a path $P(v_i,v_j)$ in EWEN $T(G)$ as defined above.

Consider the i.i.d. state process $\mathcal{M}(t)$ where each distinct state $m \in \mathcal{M}$ is indexed by the vector $(Z_i)_{v_i\in V}$. Since the number of nodes $|V|$ is finite, the total number of states in $\mathcal{M}$ is $2^{|V|}$, hence finite. Let $\pi_{\mathcal{M}}$ be the stationary distribution of $\mathcal{M}$. (Such a distribution exists because $\mathcal{M}(t)$ is i.i.d.) Let $\mathcal{K}(m)$ be the set of feasible matchings (including the null match) on bipartite graph $K(G)$ at state $\mathcal{M}(t)=m$ at time t.

Then any schedule Σ' on the EWEN induces a probability measure $\phi_m = (\phi_{mk}, k \in K(m))$ on the set of feasible matchings at state m, for all states $m \in \mathcal{M}$; such that $\phi_{mk} \geq 0$ for all $k \in \mathcal{K}(m)$ and $\sum_{k\in K(m)} \phi_{mk} = 1$. In other words, the schedule selects a convex combination of the set of all feasible matchings over all states.

Let the long term rate serviced by any path $P(v_i,v_j)$ is given by

$$\nu_{ij}(\phi) = \sum_{m\in\mathcal{M}} \pi_{\mathcal{M}}(m) \sum_{k\in K:P(v_i,v_j)\in A^{\Sigma'}(t)} \phi_{mk} R_i.$$

Then $\nu(\phi) \stackrel{\Delta}{=} (\nu_{ij}(\phi))_{i,j=1}^{|E|}$ is the service point corresponding to the induced measure ϕ.

Definition 10. *A* static split schedule ϕ_0 *is defined by a measure* ϕ_0[3]*, over the set of feasible matchings on* $K(G)$ *where, at time* t*, if the Markov chain* $\mathcal{M}(t) = m$*, then the scheduler picks matching* $k \in \mathcal{K}(m)$ *with probability* $\phi_{0,mk}$*.*

The rule is *static* in the sense that the scheduling decision depends only on the current *state* of the system.

Claim 1. *For any feasible service point for the EWEN, under the constraint that nodes* $v_i', v_i'' \in V_T$ *do not queue packets, there exists a* static split schedule ϕ *that can achieve that rate. In other words, a static split rule based schedule is maximal on the set of all feasible service rates on the EWEN.*

Proof: See Section 3.8.1. ∎

Since from the above claim there exists an SSS to achieve any feasible rate on EWEN $T(G)$, there must also be an SSS to achieve the maximum end-to-end rate on the EWEN $T(G)$ under the constraint that no $v', v'' \in V_T$ store packets. We will now characterize the maximum end-to-end rate. First, we show that for any flow $\tilde{f}$ that meets the constraints of Defintion 8, an information theoretic unicast rate of $\tilde{f}(1 - O(1/q))$ is feasible on EWEN $T(G)$.

Claim 2. *If* $\tilde{Z}_j = Z_j$*, then there exists a time-indexed collection of active path sets,* $A^{\Sigma'}(\omega, t)$ *that satisfies information theoretic unicast rate* $\tilde{f}(1-O(1/q))$ *on EWEN* $T(G)$*. Equivalently, we can find a schedule* $\Sigma'(\omega, t)$ *on EWEN* $T(G)$ *such that an innovation flow of* $\tilde{f}(1-O(1/q))$ *is achieved.*

Proof: See Section 3.8.2. ∎

Lemma 9. *If* $\tilde{Z}_j = Z_j$ *for all* $v_j \in V$*, the maximum end-to-end rate on* $T(G)$ *is given by* $\bar{C}_q(1 - O(1/q))$*.*

Proof: By Claim 2, any flow $\tilde{f}(1 - O(1/q))$ as given by the constraints in Defintion 8 is feasible on EWEN $T(G)$. Then, by Definition 8, it trivially follows that the sum-rate of flow across any cut $(\mathcal{S}, V \backslash \mathcal{S})$, where $S \subseteq V$, is given by $V_{\mathcal{T}(G)}(\mathcal{S})(1 - O(1/q))$.

[3]We will reuse the notation here, a static split schedule given by $\phi_\cdot$ will indicate that the corresponding measure is $\phi_\cdot$.

Further, by the max-flow min-cut theorem, we know that the maximum value of $\sum_{P\in\mathcal{P}} \tilde{f}_P$ corresponds to the smallest cut in the EWEN. Hence the maximum end-to-end rate on $T(G)$ is given by $\bar{C}_q(1-O(1/q))$. ∎

We will now construct a schedule Σ on BAIN G based on Σ' as follows:

(i) For each $P(v_i, v_j) \in A^{\Sigma'}(t)$, u_i transmits $X_i = RLC(S_t(v_i))$

(ii) v_j receives the symbol $Y_j = \sum_{k\in\Gamma_I(v_j)} h_{kj}X_k$;

(iii) for each v_k such that $\bigcup_{v_k\in V} P(v_k, v_l) \cap A^{\Sigma'}(t) = \emptyset$, v_k does not transmit any symbol;

(iv) for each v_k such that $\bigcup_{v_k\in V} P(v_l, v_k) \cap A^{\Sigma'}(t) = \emptyset$, then v_k drops all received packets.

Analogous to the definition in (3.24), of a path set on EWEN $T(G)$ under schedule Σ' at time t, we can define the active-edge set

$$A^{\Sigma}(t) \triangleq \{(v_i, v_j) | P(v_i, v_j) \in A^{\Sigma'}(t)\}$$

on BAIN G.

Let us introduce notation for the transmitters and receivers in $A^{\Sigma}(t)$ as follows,

$$\begin{aligned} V_{\text{tx}}^{\Sigma}(t) &\triangleq \{v_i \in V | (v_i, v_j) \in A^{\Sigma}(t)\} \\ V_{\text{rx}}^{\Sigma}(t) &\triangleq \{v_j \in V | (v_i, v_j) \in A^{\Sigma}(t)\}. \end{aligned}$$

Further, by Definition 9, all nodes in the subgraph $G(A^{\Sigma}(t))$, induced by the active set $A^{\Sigma}(t)$, have in-degree and out-degree no more than 1. Hence, we can make a 1-1 correspondence between the transmitters and receivers in $G(A^{\Sigma}(t))$.

Note that, in general, $V_{\text{tx}}^{\Sigma}(t) \cap V_{\text{rx}}^{\Sigma}(t) \neq \emptyset$ and so $(V_{\text{tx}}^{\Sigma}(t), V_{\text{rx}}^{\Sigma}(t))$ is not necessarily a cut on $G(A^{\Sigma}(t))$.

3.6.4 "Red" packets and the random event $\mathcal{D}(t)$

Consider an active set $A^{\Sigma}(t)$ in G, with transmitters $u_i \in V_{\text{tx}}^{\Sigma}(t)$ labeled $i = 1, 2, \ldots, m$ and receivers $v_j \in V_{\text{rx}}^{\Sigma}(t)$ labeled $j = 1, 2, \ldots, m$. Here, $m = |V_{\text{tx}}^{\Sigma}(t)| = |V_{\text{rx}}^{\Sigma}(t)| = |A^{\Sigma}(t)|$.

Recall that the nodes in the subgraph $G(A^{\Sigma}(t))$ have in-degree and out-degree no more than 1 and we can make a 1-1 correspondence between the transmitters and receivers in $G(A^{\Sigma}(t))$.

Let $V_0(t) \subseteq V_{\text{rx}}^{\Sigma}(t)$ be the set of nodes in the BAIN G such that the corresponding nodes in EWEN $T(G)$ receive innovations at time t. Let $U_0(t) \subseteq V_{\text{tx}}^{\Sigma}(t)$ be the corresponding set of transmitters in BAIN G. Also let $m_0(t) \triangleq |V_0(t)| = |U_0(t)|$; $m_0(t) \leq m$.

Since the nodes in $V_0(t)$ are part of a DAG, there exists a topologically ordered collection of indices $\mathcal{J} = \{j_1, j_2, \ldots, j_{m_0(t)}\}$, and the corresponding collection of nodes $V_0(t) = \{v_{j_1}, v_{j_2}, \ldots, v_{j_{m_0(t)}}\}$ along with an (unordered) index set $\mathcal{K} = \{k_1, k_2, \ldots, k_{m_0(t)}\}$, and the corresponding distinct collection of nodes $U_0(t) = \{u_{k_1}, u_{k_2}, \ldots, u_{k_{m_0(t)}}\}$, such that $(u_{k_i}, v_{j_i}) \in A^{\Sigma}(t)$. In other words, u_{k_i} transmits a packet at time t and v_{j_i} receives the $\mathbb{F}_q$-sum of all incident codewords at itself.

Let us define the row vectors

$$\bar{h}_i \triangleq (\eta_{0,j_i}, \eta_{1,j_i}, \ldots, \eta_{m_0(t),j_i})$$

where $\eta_{l,j_i} = h_{k_l,j_i}$ if $u_{k_l} \in \Gamma_I(v_{j_i})$, and 0 otherwise. Further, let

$$\mathbf{H}^t \triangleq [\bar{h}_1^* \bar{h}_2^* \ldots \bar{h}_{m_0(t)}^*]^*$$

To count the progression of innovative packets on the BAIN, we will consider a subset of packets available at any node. In the following, we will denote this set $S_t^R(v_j)$ of packets as "red" packets and the rest as "black" packets; thereby partitioning the set of packets at node $v_j \in V$ at any time t as follows

$$S_t(v_j) = S_t^R(v_j) \cup S_t^B(v_j) \tag{3.25}$$

where $S_t^R(v_j) \cap S_t^B(v_j) = \emptyset$.

Definition 11. *We will define the set $S_t^R(v_j)$ by construction as follows:*
Initialize: *$S_0^R(v_1) = \{a_1, a_2, \ldots, a_K\}$, $S_0^R(v) = \emptyset$ for all $v \in V \backslash v_1$.*
Recursion *Let $S_t^R(v)$ be the sets of "red" packets at nodes $v \in V$ at time $t \in \mathbb{N}$ such that $\sum_{v \in V} |S_t^R(v)| = K$, and the set has span K. This is trivially true at $t = 0$.*

Let A_t^{Σ} be the schedule at time t and the sets $U_0(t)$ and $V_0(t)$ are as defined above. We will also use the indexing $u_{k_i} \in U_0(t)$, $v_{j_i} \in V_0(t)$ corresponding to schedule Σ in time t. Let us index the total set of red packets as

$$S_t^R(V) = \{\hat{x}_1^t, \hat{x}_2^t, \ldots, \hat{x}_{m_0(t)}^t, \hat{x}_{m_0(t)+1}^t, \ldots, \hat{x}_K^t\}$$

where each of the first $m_0(t)$ red packets satisfy that $\hat{x}_i^t \in S_t^R(u_{k_i})$ for $i = 1, 2, \ldots, m_0(t)$ and the rest of the red packets are indexed arbitrarily.

If $S_t^R(V)$ has span K, and since all packets are formed by linear combinations of the original message set, we can express each packet $x \in S_t(u_{k_i})$ as a linear combination of the spanning set $S_t^R(V)$ as follows

$$x = \sum_{j=1}^{K} \xi_j^t(x)\hat{x}_j^t$$

where $\xi_j^t \in \mathbb{F}_q$ are the corresponding coefficients. Note that this applies to both "red" and "black" packets. Further, note that since $\hat{x}_i^t$, for $i = 1, 2, \ldots, m_0$ is present in node u_{k_i}, the coefficient

$$\xi_i^t(\hat{x}_i^t) = 1. \tag{3.26}$$

Then the codeword X_i^t transmitted by each u_{k_i} can be expressed as

$$\begin{aligned} X_i^t &\triangleq \sum_{x \in S_t(v)} \beta_{i,v}^t x \\ &= \sum_{x \in S_t(v)} \beta_{i,v}^t \sum_{j=1}^{K} \xi_j^t(x)\hat{x}_j^t \qquad (3.27) \\ &= \sum_{j=1}^{K} \beta_{i,v}^t \Big(\sum_{x \in S_t(v)} \xi_j^t(x) \Big) \hat{x}_j^t \qquad (3.28) \\ &= \sum_{j=1}^{K} \nu_{ij}^t \hat{x}_j^t. \qquad (3.29) \end{aligned}$$

Let the column vector

$$\nu_{i,1}^t \triangleq [(\nu_{ij}^t)_{j=1}^{m_0}]^*; \nu_{i,2}^t \triangleq [(\nu_{ij}^t)_{j=m_0+1}^{K}]^*$$

where $\mathbf{a}^*$ denotes the transpose of vector $\mathbf{a}$.

Let $\mathcal{B}^t$ denote the event that the coefficient vectors $\{\nu_{i,1}^t\}_{i=1}^{m_0(t)}$ are linearly independent.

Let $Y_i^t \triangleq \sum_{l \in \Gamma_I(v_{j_i})} h_{k_l,j_i} X_i^t$ be the symbol received at each v_{j_i}. For all $i = 1, 2, \ldots, m_0(t)$, let us define events

$$\mathcal{A}_i^t \triangleq \{\nu_{ii} \neq 0\},$$
$$\mathcal{H}^t \triangleq \left\{\mathbf{H}^t \text{ is full rank}\right\}.$$

We can then define the event

$$\mathcal{D}(t) \triangleq \mathcal{B}^t \cap \mathcal{H}^t \bigcap_{i=1,2,\ldots,m_0(t)} \mathcal{A}_i^t. \tag{3.30}$$

Then we can define the sets $S^R_{t+1}(v)$ as follows:

(1) If $\mathcal{D}(t) \cap (S^R_t(V)$ has span $K)$[4]

1a. For each $u_{k_i} \in U_0(t)$, pick any $x_l \in S^R_t(u_{k_i})$ such that the corresponding $\beta_{l,new} \neq 0$ and change x_l from a "red" packet to a "black" packet; thus $S^B_{t+1}(u_{k_i}) \stackrel{\Delta}{=} S^B_t(u_{k_i}) \cup \{x_l\}$ and $S^R_{t+1}(u_{k_i}) \stackrel{\Delta}{=} S^R_t(u_{k_i}) \backslash \{x_l\}$.

1b. For each $v_{j_i} \in V_0(t)$, $S^R_{t+1}(v_{j_i}) \stackrel{\Delta}{=} S^R_t(v_{j_i}) \cup \{Y^t_i\}$.

(2) Else,

2a. No change happens to the packets in $U_0(t)$, i.e. $S^R_{t+1}(u_{k_i}) = S^R_t(u_{k_i})$ for all $u_{k_i} \in V_0(t)$.

2b. For each $v_{j_i} \in V_0(t)$, $S^B_{t+1}(v_{j_i}) \stackrel{\Delta}{=} S^B_t(v_{j_i}) \cup \{Y^t_i\}$.

To prove that the above algorithm to mark "red" packets progresses in time, we first show that if $\mathcal{D}(t)$ occurs, and the set of red packets at time t, $S^R_t(V)$ is full rank, then the set of red packets at time $t+1$ is full rank as well.

Lemma 10. *($S^R_t(V)$ has span K)* $\implies$ *($S^R_{t+1}(V)$ has span K)*

Proof: It is trivial to check by the construction above that if $\mathcal{D}(t)$ fails, then $S^R_{t+1}(V) = S^R_t(V)$, the set of red packets are not updated; hence if $S^R_t(V)$ has span K) then $S^R_{t+1}(V)$ has span K).

If condition $\mathcal{D}(t)$ is true, the set of red packets at time $t+1$ is, using the notation from Definition 11,

$$S^R_{t+1}(V) = \{Y^t_1, Y^t_2, \ldots, Y^t_{m_0(t)}, \hat{x}^t_{m_0(t)+1}, \ldots, \hat{x}^t_K\}$$

[4]We are using the same notation for a logical condition and an event. If A is a logical condition, the event is actually $\mathcal{A} = \{\omega | A(\omega) = T\}$

Let

$$\begin{aligned}
\mathbf{Y}^t &\triangleq [Y_1^t Y_2^t \dots Y_{m_0(t)}^t]^* \\
\mathbf{X}^t &\triangleq [X_1^t X_2^t \dots X_{m_0(t)}^t]^* \\
\mathbf{N}_1^t &\triangleq [\nu_{1,1}^t \nu_{2,1}^t \dots \nu_{m_0(t),1}^t]^* \\
\mathbf{N}_2^t &\triangleq [\nu_{1,2}^t \nu_{2,2}^t \dots \nu_{m_0(t),2}^t]^* \\
\hat{\mathbf{X}}_1^t &\triangleq [\hat{x}_1^t \hat{x}_2^t \dots \hat{x}_{m_0(t)}^t] \\
\hat{\mathbf{X}}_2^t &\triangleq [\hat{x}_{m_0(t)+1}^t \hat{x}_{m_0(t)+2}^t \dots \hat{x}_K^t] \\
\hat{\mathbf{X}}^t &\triangleq [\hat{\mathbf{X}_1}^t | \hat{\mathbf{X}_1}^t]
\end{aligned}$$

Then, we may write,

$$\begin{aligned}
\mathbf{Y}^t &= \mathbf{H}^t \mathbf{X}^t \\
&= \mathbf{H}^t [\mathbf{N}_1^t | \mathbf{N}_2^t] [\hat{\mathbf{X}_1}^t | \hat{\mathbf{X}_2}^t]^* \\
&= [\mathbf{H}^t \mathbf{N}_1^t | \mathbf{H}^t \mathbf{N}_2^t] [\hat{\mathbf{X}_1}^t | \hat{\mathbf{X}_2}^t]^* \\
&= \mathbf{G}^t \hat{\mathbf{X}}^{t*}
\end{aligned}$$

where $\mathbf{G}^t \triangleq \{\gamma_{ij}^t\}_{i=1,j=1}^{i=m_0(t),j=K}$ satisfying

$$Y_i^t = \sum_{j=1}^{K} \gamma_{ij}^t \hat{x}_j^t.$$

Since $\mathcal{H}^t$ guarantees that the transfer matrix $\mathbf{H}^t$ is full rank, and $\mathcal{B}^t$ implies that the coefficient matrix $\mathbf{N}_1^t$ is full rank, it implies that $\mathbf{H}^t \mathbf{N}_1^t$ is full rank. This implies that the row rank of the matrix $\mathbf{G}^t$ is m_0. It is immediate, then, that the set $\{Y_1^t, Y_2^t, \dots, Y_{m_0(t)}^t\}$ is linearly independent, i.e. with rank $m_0(t)$.

Further, since $\mathbf{H}^t \mathbf{N}_1^t$ is full rank, none of the rows of $\mathbf{H}^t \mathbf{N}_1^t$ are all zero. Hence, for each $i < m_0(t)$, there exists at least one $l \in 1, 2, \dots, m_0(t)$ such that coefficient $\gamma_{il}^t \neq 0$. Thus, for each $i < m_0(t)$ $Y_i^t \notin \text{span}(\{\hat{x}_{m_0(t)+1}^t, \dots, \hat{x}_K^t\})$.

It follows that $S_{t+1}^R(V)$ has rank K. ∎

3.6.5 Counting Innovations on the BAIN G

Recall that in the previous section, we relied upon disparate sets to store and track the progression of innovative packets over a wireless erasure network; correspondingly we

defined the stored innovation for path P on node $v_j \in V$ at time-slot t by the queue relation $Q_t^P(v_j) = \text{span}(S_t^P(v_k)) - \text{span}(S_t^P(v_j))$, where $(v_j, v_k) \in P$.

However, on account of the broadcast and additive interference nature of the wireless transmitter and receiver respectively, it is not possible to separate packets on various paths in a BAIN.

Lemma 11. *span*$(S_t^R(v_j)) \subseteq$ *span*$(S_t(v_j))$

Proof: Follows from the partitioning in (3.25). ∎

Thus, to show that the BAIN achieves a rate of f, it suffices to show that

$$\lim_{\Delta \to \infty} \inf \frac{1}{\Delta} |\text{span}(S_\Delta^R(v_n))| = f.$$

Remark 9. *The distinction between red and black packets are only for the purpose of analysis and to be able to bound the size of the span of the set of codeword vectors at the destination v_n. However, the coding algorithm does not distinguish between black and red packets and considers the set of all packets $S_t(v_j)$ to form the RLC at time t.*

3.6.6 Coupling Z_j on BAIN with $\tilde{Z}_j$ on the EWEN

Let us consider the probability space $(\Omega', \mathcal{F}, \mathbb{P})$ over which a rate f of flow of innovative packets is possible on the EWEN $T(G)$; similarly, let $(\Omega, \mathcal{F}, \mathbb{P})$ be the probability space of the events in BAIN G. We will couple the sample path $\omega \in \Omega$ for the BAIN G with $\omega' \in \Omega'$ for EWEN $T(G)$ as follows.

(i) $\tilde{Z}_{i,t}(\omega') = 1$ in $T(G)$ if $\{Z_{i,t}(\omega) = 1\} \cup \{\mathcal{D}(t) = 1\} \bigcup_{v \in \Gamma_I(v_i)} \{\mathcal{A}_v^t = 0\}$ on BAIN G

(ii) $R_{i,t}(\omega') = 1$ in $T(G)$ if $R_{i,t}(\omega) = 1$ on G.

Lemma 12. *At any time t, let $Q_i^{\Sigma'}(t)$ be the set of queue sizes on $T(G)$ under schedule Σ'. Then under the sample path coupling constructed above, $|S_t^R(v_i, \omega)| = Q_i^{\Sigma'}(t, \omega')$ for all $i = 2, 3, \ldots, n$, where $n = |V|$.*

Proof: Trivially, by the construction of the set of red sets in Definition 11 and the coupling above. ∎

Let $\mathcal{F}_t$ be the filtration sequenced against time t defined on the probability space Ω.

Lemma 13. $\mathbb{P}(\mathcal{D}^t|\mathcal{F}_t) = 1 - O(1/q)$.

Proof:

$$\begin{aligned}\mathbb{P}(\mathcal{D}^t|\mathcal{F}_t) &= \mathbb{P}(\mathcal{B}^t \cap \mathcal{H}^t \bigcap_{i=1,2,\ldots,m_0(t)} \mathcal{A}_i^t|\mathcal{F}_t) \\ &= \mathbb{P}(\bigcap_{i\leq m_0(t)} \mathcal{A}_i^t|\mathcal{F}_t)\mathbb{P}(\mathcal{B}^t|\bigcap_{i\leq m_0(t)} \mathcal{A}_i^t\mathcal{F}_t)\mathbb{P}(\mathcal{H}^t|\mathcal{F}_t) \qquad (3.31)\end{aligned}$$

where the last relation follows, since the channel matrix $\mathbf{H}^t$ is independent of the transmitted codewords X_i^t.

Note that in (3.29), the coefficient of $\hat{x}_i^t$, $\nu_{ii}^t = \beta_{i,v}^t(\sum_{x\in S_t(v)} \xi_j^t(x))$, where the $\beta_{i,v}$ are random coefficients in $\mathbb{F}_q$ and $\xi_j^t(x)$ are as defined in (3.27).

Further, from Lemma 12 we see that conditioned on $\mathcal{F}_t$, each of the nodes in $u_{k_i} \in U_0$ have at least one "red" packet, which is indexed as $\hat{x}_i^t$. Now, since the packet $\hat{x}_i^t \in S_t(u_{k_i})$ is "red" in node u_{k_i}, this implies that $\xi_i^t(x) = 1$ for the particular instance of $x = \hat{x}_i^t$. Thus, we can write

$$\sum_{x\in S_t(v)} \xi_i^t(x) = 1 + \sum_{x\in S_t(v)\setminus\hat{x}_i^t} \xi_i^t(x)$$

Thus,

$$\begin{aligned}& \mathbb{P}(\mathcal{A}_i^t|\mathcal{F}_t) \\ = \;& \mathbb{P}(\nu_{ii}^t \neq 0|\mathcal{F}_t) \\ \overset{(a)}{=} \;& \mathbb{P}(\beta_{i,v}^t(\sum_{x\in S_t(v)} \xi_i^t(x)) \neq 0|\mathcal{F}_t) \\ \overset{(b)}{=} \;& \mathbb{P}(\beta_{i,v}^t \neq 0|\mathcal{F}_t)\mathbb{P}\left(\sum_{x\in S_t(v)\setminus\hat{x}_i^t} \xi_i^t(x) \neq q-1|\mathcal{F}_t\right) \\ = \;& (1-1/q)(1-1/q) \\ \geq \;& 1 - 2/q. \qquad (3.32)\end{aligned}$$

where (a) is from the definition of ν_{ij}^t in Equation 3.29, and (b) follows since the coefficient $\xi_i^t(\hat{x}_i^t) = 1$ from Equation (3.26).

Consider now, the case where for any pair of distinct nodes $u_{k_i}, u_{k_{i'}} \in U_0$, $\nu_{ii}^t \neq \nu_{i'i'}^t$. If this condition is satisfied, clearly all the coefficient vectors $\{\nu_{i,1}^t\}_{i=1}^{m_0(t)}$ are linearly independent. Thus,

$$\{\nu_{jj}^t|\nu_{ii}^t \neq \nu_{i'i'}^t \forall u_{k_i}, u_{k_{i'}} \in U_0; u_{k_i} \neq u_{k_{i'}}\} \subseteq \mathcal{B}_t$$

Equivalently,

$$\mathbb{P}(\{\nu_{jj}^t|\nu_{ii}^t \neq \nu_{i'i'}^t \forall u_{k_i}, u_{k_{i'}} \in U_0; u_{k_i} \neq u_{k_{i'}}\}| \bigcap_{i\leq m_0(t)} \mathcal{A}_i^t\mathcal{F}_t)$$
$$\leq P(\mathcal{B}_t| \bigcap_{i\leq m_0(t)} \mathcal{A}_i^t\mathcal{F}_t). \tag{3.33}$$

Now, conditioned on event $\mathcal{A}_i^t$, $\beta_{i,v}^t \neq 0$. Therefore, since $\beta_{i,v}^t$ is chosen uniformly at random from $\mathbb{F}_q$, conditioned on $\mathcal{A}_i^t$, the coefficients $\beta_{i,v}^t \in \mathbb{F}_q\backslash\{0\}$ are uniformly distributed over $\mathbb{F}_q\backslash\{0\}$. Also, conditioned on $\mathcal{A}_i^t$, $\mu \stackrel{\Delta}{=} (\sum_{x\in S_t(v)} \xi_j^t(x)) \neq 0$; therefore $\mu\beta_{i,v}^t$ is uniformly distributed over $\mathbb{F}_q\backslash\{0\}$.

Thus, conditioned on $\mathcal{A}_i^t$, ν_{ii}^t is uniformly distributed over $\mathbb{F}_q\backslash\{0\}$. Hence, from (3.33),

$$P(\mathcal{B}_t| \bigcap_{i\leq m_0(t)} \mathcal{A}_i^t\mathcal{F}_t)$$
$$\geq \mathbb{P}(\{\nu_{jj}^t|\nu_{ii}^t \neq \nu_{i'i'}^t \forall u_{k_i}, u_{k_{i'}} \in U_0; u_{k_i} \neq u_{k_{i'}}\}| \bigcap_{i\leq m_0(t)} \mathcal{A}_i^t\mathcal{F}_t)$$
$$= 1 - (\frac{1}{q-1})^{m_0(t)-1}$$
$$\geq 1 - \kappa_1(1/q). \tag{3.34}$$

where $\kappa_1 \stackrel{\Delta}{=} (m_0 - 1)\frac{q}{q-1}$ is a positive constant less than $|V|$.

Finally, observe that the elements in the channel vector $\bar{h}_i$ are each chosen uniformly at random from $\mathbb{F}_q$. Hence from [18], it follows that

$$\mathbb{P}(\mathcal{H}_t|\mathcal{F}_t) = 1 - O(1/q). \tag{3.35}$$

Plugging the relations in (3.32), (3.34) and (3.35) in the R.H.S. of (3.31), we are done. ∎

Hence at any time t, the size of the span of the red packets at the destination v_n in the BAIN G is the same as the size of the span of packets at v_n in the EWEN $T(G)$. Since by Lemma 11, the span of the "red" packets is less than the span of all packets at the destination v_n in the BAIN, this also implies that the span of all packets at the destination v_n in the BAIN is greater than equal to the span of all packets at the destination in the EWEN $T(G)$. Hence, the BAIN achieves a rate equal to the max-flow min-cut rate for EWEN $T(G)$.

The last piece of the puzzle is in showing that the correlated drop process $\mathcal{D}_t$ does not reduce the rate by more than a $O(1/q)$ fraction. To do so, we will use the bound in Lemma 13 as follows.

Let us define the random variable $\xi_t(\omega) \stackrel{\Delta}{=} 1$ when $\omega \in (\mathcal{D}^t)^c$ and $\xi_t(\omega) \stackrel{\Delta}{=} 0$, otherwise. In Lemma 13, let K_d be the constant such that $\mathbb{P}(\mathcal{D}^t|\mathcal{F}_t) \geq 1 - (K_d/q)$. Let us also define the random variables $\hat{Z}_t \sim \text{Bernoulli}(K_d/q)$.

Lemma 14. *Uniformly over all $T \geq 1$,*

$$0 \leq \sum_{t=1}^{T} \xi_t \leq_{st} \sum_{t=1}^{T} \hat{Z}_t.$$

Proof: We will induce over T. At $T = 1$, $\mathcal{F}_t = \Omega$ and it is trivially true from Lemma 13 that $\mathbb{P}(\xi_1 = 1|\mathcal{F}_1) \leq K_d/q = \mathbb{P}(Z_1 = 1)$.

Let us assume that the induction hypothesis holds at $T-1$, i.e. $\sum_{t=1}^{T-1} \xi_t \leq_{st} \sum_{t=1}^{T-1} Z_t$. We will now show that the induction hypothesis holds at T.

For any $\alpha > 0$, we can write

$$\begin{aligned}
& \mathbb{P}(\sum_{t=1}^{T} \xi_t > \alpha) \\
= \; & \mathbb{P}(\{\xi_T = 1\}|\sum_{t=1}^{T-1} \xi_t > \alpha - 1)\mathbb{P}(\sum_{t=1}^{T-1} \xi_t > \alpha - 1) \\
& + \mathbb{P}(\{\xi_T \geq 0\}|\sum_{t=1}^{T-1} \xi_t > \alpha)\mathbb{P}(\sum_{t=1}^{T-1} \xi_t > \alpha) \\
\\
= \; & \mathbb{P}(\{\xi_T = 1\}|\sum_{t=1}^{T-1} \xi_t > \alpha - 1)\mathbb{P}(\sum_{t=1}^{T-1} \xi_t > \alpha - 1) \\
& + \mathbb{P}(\sum_{t=1}^{T-1} \xi_t > \alpha)
\end{aligned}$$

where the last relation follows since the random variables ξ_t are non-negative.

It is immediate from Lemma 13 that for any time-slot t, $\mathbb{P}(\xi_t = 1|\mathcal{F}_t) \leq K_d/q = \mathbb{P}(\hat{Z}_t = 1)$. Thus,

$$\begin{aligned} & \mathbb{P}(\sum_{t=1}^{T} \xi_t > \alpha) \\ \leq \quad & \mathbb{P}(\hat{Z}_T = 1)\mathbb{P}(\sum_{t=1}^{T-1} \hat{Z}_t > \alpha - 1) + \mathbb{P}(\sum_{t=1}^{T-1} \hat{Z}_t > \alpha) \\ = \quad & \mathbb{P}(\sum_{t=1}^{T} \hat{Z}_t > \alpha). \end{aligned}$$

∎

We are now ready to state the main achievability result.

Theorem 4. *A rate of $\bar{C}_q - O(1/q)$ is achievable on the BAIN G.*

Proof: Let us denote the number of dropped packets at time t due to the event $\xi_t = 1$ by the random variable L_t. Since, by Claim 2, the total number of packets scheduled over the network cannot exceed $|V|$[5], we can bound $L_t(\omega) \leq |V|\xi_t(\omega)$.

Then, from Lemma 14 above, it immediately follows that

$$\limsup \frac{1}{T} \sum_{t=1}^{T} L_t(\omega) \leq \lim \frac{|V|}{T} \sum_{t=1}^{T} Z_t = \frac{|V|K_d}{q}.$$

The above relation states that the rate loss in the EWEN H due to the correlated dropping process $\{\mathcal{D}_t\}$ is bounded by a fraction $O(1/q)$. Recall that by Lemma 9, we know that if $\tilde{Z}_i = Z_i$, the flow rate of innovation across EWEN $T(G)$ is $\bar{C}_q(1 - O(1/q))$. Hence, flow rate achieved by the EWEN $T(G)$, when $P(\mathcal{D}_t) = O(1/q)$ is $\bar{C}_q(1 - O(1/q)) - O(1/q)$. By Lemma 12, then, the flow rate of "red" packets at the destination node v_d in the BAIN G is the same as the flow of packets at v_d in EWEN $T(G)$. Further, by the subset property in Lemma 11, this implies that a unicast innovation flow rate of $\bar{C}_q(1 - O(1/q)) - O(1/q)$ is achievable on the BAIN G.

This readily implies the result. ∎

[5]Recall that the nodes are full duplex

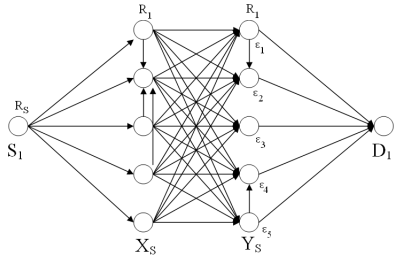

Figure 3.4: Capacity across the cut in the DAG above, $R_S = 10R_1 < \log q$ Nodes are labelled with erasure probabilities ϵ_i.

3.7 Capacity Gain due to Fading

We illustrate the gain in network capacity due to fading diversity by analyzing the capacity of the heterogenous network given in Figure 3.4 under fading and non-fading cases. Specifically, we compare the unicast capacity from S_1 to D_1 under fading with an upper bound for the non-fading case.

The source S_1 is connected to each of its outgoing nodes by wireline links of rate R_1, the nodes on the left edge of the cut $(S, \bar{S})$ transmit over wireless links to the nodes on the cut's right edge, and the latter transmit to D_1 over wireline links, each of rate R_1. Let us label the nodes on the left edge of the cut as $u_i, i = 1, 2, \ldots 5$ and the nodes on the right edge of the cut as $v_i, i = 1, 2, \ldots, 5$.

Suppose that R_1 and q are such that the cut $(S, \bar{S})$ is the bottleneck cut (for instance, $R_1 = \log q$). Then, from Theorem 4, the capacity of the unicast from S_1 to D_1 under uniform i.i.d. fading is $R_1 \sum_{i=1}^{5}(1 - \epsilon_i)(1 - O(1/q))$.

In contrast, if the links crossing the cut have no fading, each of the nodes on the right hand side of the cut $(S, \bar{S})$ receive the same symbol in case there is no erasure at the corresponding receiver. In other words, if X_i is the message symbol transmitted by each node u_i on the left hand side of the cut, each node v_i on the right hand side of the cut receives the symbol $\sum_{j=1}^{5} X_j$ with probability $1 - \epsilon_i$ and the erasure symbol $\mathcal{E}$ with probability ϵ_i.

Direct computation yields that the upper bound on the capacity of this cut is $R_1(1 - \prod_{j=1}^{5} \epsilon_i)$.

Thus, for small erasure probabilities, approximately 5-fold increase in the capacity is afforded by fading diversity in the example network. Clearly, gains will be higher for

graphs with larger bottleneck bipartite subgraphs embedded in them.

3.8 Proofs

3.8.1 Proof of Claim 1

Pick any feasible service point $\mu \stackrel{\Delta}{=} (\mu_{e_i})_{i=1}^{|E|}$. Then, since this service point is feasible under the constraint that the nodes in V_T do not queue packets (see Remark 5), the entire path $P(v_i, v_j)$ must be scheduled at the same time. Further, this also implies that paths $P(v_i, v_j)$ and $P(v_i, v_k)$ or $P(v_k, v_j)$ cannot be scheduled at the same time.

In particular, let $\xi(\mu)$ be a schedule that achieves the rate point μ. Also, let us fix some large interval of time $0, 1, \ldots, T$. Let $\hat{\phi}^{\xi}_{mk}$ be the fraction of time-slots that $\mathcal{M}(t) = m$ and matching $k \in \mathcal{K}(m)$ is chosen on bipartite graph $K(G)$ under schedule $\xi(\mu)$:,

$$\hat{\phi}^{\xi}_{mk} \stackrel{\Delta}{=} \frac{1}{T} \sum_{t=1}^{T} \mathbf{1}\{\mathcal{M}(t) = m \cap k \in \mathcal{K}(m) \text{is chosen}\} \tag{3.36}$$

Then, trivially,

$$\sum_{m \in \mathcal{M}} \sum_{k \in \mathcal{K}(m)} \hat{\phi}^{\xi}_{mk} = 1.$$

Further, since service point μ is achieved by schedule ξ,

$$\mu_{ij} = \sum_{m \in \mathcal{M}} \sum_{\{k \in \mathcal{K}(m,i,j)\}} \hat{\phi}^{\xi}_{mk} R_i$$

where $\mathcal{K}(m, i, j) \stackrel{\Delta}{=} \{k \in \mathcal{K}(m) : (v_i, v_j) \in Tx(V) \times Rx(V)\}$.

Now define a static split schedule ϕ^{SSS} as follows

$$\phi^{SSS}_{mk} = \hat{\phi}^{\xi}_{mk} / \pi_{\mathcal{M}}(m). \tag{3.37}$$

where $\mathbb{P}(\mathcal{M}(t) = m) = \pi_{\mathcal{M}}(m)$.

Then, we can write the empirical rate of innovative flow through $P(v_i, v_j)$ under schedule ϕ^{SSS} as

$$\mu_{ij}(T) = \frac{1}{T} \sum_{m \in \mathcal{M}} \sum_{k \in \mathcal{K}(m,i,j)} \mathbf{1}\{\mathcal{M}(t) = m \cap k \in \mathcal{K}(m) \text{is chosen}\} R_i$$

Let $T_m \stackrel{\Delta}{=} |\{t : \mathcal{M}(t) = m\}|$. By the Strong law of large numbers, $\frac{T_m}{T} \to \pi_{\mathcal{M}}(m)$ as $T \to \infty$. Also, since the SSS picks matching $k \in \mathcal{K}(m)$ i.i.d. with probability ϕ^{SSS}_{mk}, by ergodicity, we have that $\frac{1}{T_m} \mathbf{1}\{\mathcal{M}(t) = m \cap k \in \mathcal{K}(m) \text{is chosen}\} \to \phi^{\xi}_{mk}$ as $T \to \infty$.

Thus, $\mu_{jk}(T) \to \mu_{jk}$ as $T \to \infty$. ∎

3.8.2 Proof of Claim 2

To begin with, recall that $\tilde{Z}_j = Z_j$ denotes that the edge erasure process at (v_j'', v_j) on EWEN $T(G)$ is identical to the node erasure process Z_j at node $v_j \in V$ on BAIN G.

Observe that according to schedule Σ' [see Equation (3.24)], a packet traverses path $P(v_i, v_j)$ at time t, only if $Q_i(t) > 0$ and $Z_j = 0$. Hence, the condition in Definition 9 that if $P(v_i, v_j) \in A^{\Sigma'}(t)$, then $P(v_k, v_j), P(v_i, v_k) \notin A^{\Sigma'}(t)$ for all $v_k \neq v_i, v_j$ at time t implies that $A^{\Sigma'}(\omega, t)$ is a matching on the set of transmitters with receivers where there is no erasure.

Let, $\tilde{\mu}_{ij}$ be the flow rate achievable between the transmitter $Tx(v_i)$ and $Rx(v_j)$,in $K(G)$.

Observe now that each $Rx(v_j)$ is scheduled only when $Z_j = 1$, with probability $1 - \epsilon_j$. Thus $\{\tilde{\mu}_{ij}\}$ must satisfy,

$$\sum_{v_i \in \Gamma_I(v_j)} \tilde{\mu}_{ij} \leq (1 - \epsilon_j) \log q.$$

Also, consider any transmitter $Tx(v_i)$. Since each path $P(v_i, v_j)$ cannot be scheduled in Σ when $Z_j = 0$, this implies that the transmitter $Tx(v_i)$ is scheduled only when at least one of the outgoing receivers does not have an erasure, i.e. with probability $(1 - \prod_{v_j \in \Gamma_O(v_i)} \epsilon_j)$. Thus $\{\tilde{\mu}_{ij}\}$ must satisfy,

$$\sum_{v_j \in \Gamma_O(v_i)} \tilde{\mu}_{ij} \leq (1 - \prod_{v_j \in \Gamma_O(v_i)} \epsilon_j) R_i.$$

Further, for conservation of flow rates, the set of edge rates $\{\tilde{\mu}_{ij}\}$ must satisfy

$$\sum_{v_j \in \Gamma_O(v_i)} \tilde{\mu}_{ij} = \sum_{v_j \in \Gamma_I(v_i)} \tilde{\mu}_{ji}$$

By Definition 8, $\tilde{f}$ satisfies the above constraints: to check this, set

$$\tilde{\mu}_{ij} \stackrel{\Delta}{=} \sum_{P : P(v_i, v_j) \in P} \tilde{f}_P.$$

Since $\tilde{f}$ is a feasible flow across the network, by Claim 1, there must be a SSS, say $\tilde{\phi}$, that achieves the flow rate.

Now, since $\mathcal{M}(t)$ is an i.i.d. process, and by the SSS $\tilde{\phi}$, matching $k \in \mathcal{K}(m)$ is picked i.i.d. conditioned on $\mathcal{M}(t) = m$, the process $\mathbf{1}\mathcal{M}(t) = m \cap k \in \mathcal{K}(m)$ is an i.i.d. process with probability $\mathbb{P}(\mathbf{1}\mathcal{M}(t) = m \cap k \in \mathcal{K}(m)) = \pi_m \tilde{\phi}_{mk}$

Analogous to (3.17), we can now define the Bernoulli process $\{\psi_{ij}(t)\}$ corresponding to each edge $(v_i, v_j) \in E$ and each time-slot t, where $\psi_{ij}(t) \in \{P' | P' \in \mathcal{P}_{ij}\}$ with probability

$$\mathbb{P}(\psi_{ij}(t) = P) = \frac{f_P}{\sum_{P' \in \mathcal{P}_{ij}} f_{P'}} \sum_{m \in \mathcal{M}} \sum_{k \in \mathcal{K}(m,i,j)} \pi_m \tilde{\phi}_{mk}. \tag{3.38}$$

Parallel to the construction in Section V-B, we can construct a sample path coupling between Ω_{ij} and Ω_{ij}^P, and accordingly, we can present a set of $|\mathcal{P}|$ path specific Skorohod problems similar to Section V-C.

Then by Lemma 3, flow $\tilde{f}(1 - O(1/q))$ is a feasible flow on EWEN $T(G)$ under the constraint that nodes in V_T do not queue packets. ∎

Chapter 4

Buffer asymptotics for coding over networks

4.1 Introduction

Network coding at intermediate routers in a network (as opposed to switching/routing) was originally proposed with a view of increasing end-to-end throughput in networks [4] and [5]. Network codes have been shown to be throughput optimal (network-wide capacity achieving) for a multicast network by Cai et. al. Furthermore, network coding via Random Linear Coding (RLC) improves network reliability and simplifies network management [8], as well as allows exploiting correlation in sensor data to improve network efficiency [9]. Recent formulations of convex optimization problems [12],[13] to characterize the sum-cost of flows through a network using RLC pose significant reduction of network-wide sum-cost for coding as opposed to routing.

Random Linear Codes applied at intermediate routers effectively spread the information from one flow across multiple flows and hence work well as an error/erasure control scheme. This spreading of information makes RLCs attractive in cases where packet drops or losses are likely to occur, such as in data dissemination over large peer-to-peer networks [18],[19],[20]. Recently, Avalanche [19] has been proposed as an alternative to BitTorrent[36] by using network codes for P2P data dissemination. Further, network codes have been proposed as a means of distributed information dispersal and recovery in large ad-hoc networks via a rumour-spreading(epidemic) model [18],[20].

The common underlying theme in much of the above work has been that network codes, and specifically RLC's, allow spatial (across the network) stochastic multiplexing across different flows and this feature can be utilized in improving reliability in large networks. Recently however, Lun, Medard and Effros [17],[21] exploit network codes for a capacity-approaching scheme for unicasts or multicasts over large networks. In their scheme, routers perform RLC over packets from different flows as well as over packets transmitted in previous time-slots. Further, for the case of Poisson traffic with i.i.d. losses at intermediate router queues (modeled as M/M/1 queues), they derive the error exponent in the large-delay regime. This is analogous to the use of block codes or convolutional codes for error control

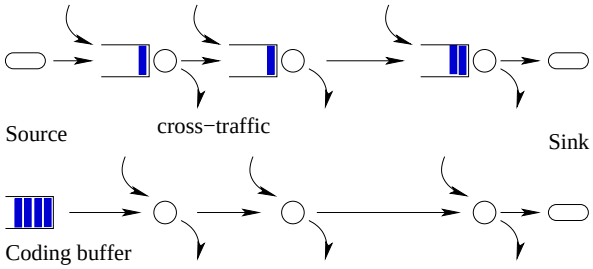

Figure 4.1: Buffering at the source versus buffering at nodes: By using network coding, a form of spatial multiplexing gain can be achieved whereby the small buffers at the nodes can be shared across multiple nodes.

in the PHY that spread the information across multiple bits in a block or neighbourhood around each bit. On a related note, error exponents of codes over networks have also been studied by Luby et. al. [37] for rateless codes [40].

The insight from [17] that packets dropped in a particular time-slots can be recovered from RLCs containing the dropped packets in future time-slots motivates us to consider the following questions:

- Can we eliminate buffering at intermediate nodes in favour of coding only at the ends? We consider the scenario where intermediate routers perform no RLCs or buffering, but merely drop packets if the link capacity is exceeded.
- Further, in the event of finite delays, how does network coding at the ends compare with queueing in intermediate routers? Here, we wish to compare QoS parameters such as delay and end-to-end packet loss probability (reliability) with coding as opposed to queueing.

We examine these questions by considering two complementary scenarios: first for the case of large networks with many flows through each node with finite buffer sizes at the sources (a many sources analysis), and second for the case of a network will a small number of flows with large buffers at the sources.

4.1.1 Large Networks: Finite Source Buffers

The main idea stems from the fact that in a very large network with N nodes and $N/2$ unicasts from each source matched to its (randomly chosen) destination, each link in the

network carries a large number of flows, say $n = \Omega(N^\alpha)$ for some $\alpha \in (0,1)$[45],[47],[48],[49]. Naturally, to ensure that the per-flow capacity on each link/edge is an $\Theta(1)$ quantity[1], the aggregate link capacity must scale with n. Stability requirements also enforce the condition that the link capacity should be greater than the mean packet arrival rate at each link. Under these conditions, we have from Chernoff's bound that the probability a link overflows is roughly of the order of $\exp(-N^\alpha \epsilon_0)$ for some $\epsilon_0 > 0$. Assuming good mixing, the probability of a link overflowing anywhere along a path of length $\Omega(N^\beta)$ for some $\beta \in (0,1)$ is approximately $O(N^\beta \exp(N^\alpha))$ which is asymptotically close to $O(\exp(N^\alpha))$ for large N. This can be interpreted as follows – the probability that there is an overflow in a single link is of the same order as the probability that there is an overflow in a path containing a polynomial number of such links. In other words, *"if an overflow occurs in a path, it will very likely occur only at only one link in the path"*. Hence, instead of buffering at each link in the path, it should suffice to buffer only at one link – translating to huge savings in buffer required per-flow for large networks and better scalability in the design of large multi-hop networks. However, the link where the overflow occurs is a function of the sample path of the arrival processes and varies with time. This variation makes it impossible to effectively *multiplex buffers across links on a path for a single flow* using traditional static buffer allocation at each link. Note that this is very different from traditional buffer multiplexing where many flows incident at a single link share buffers across flows [50],[51], [52].

It is in this scenario that we propose network coding as a means of "sharing" memory across links along a flow path. We call this *spatial buffer multiplexing* – where buffering and coding implemented via a sliding window of packets at the source compensates for packet loss at any downstream bufferless link. In addition to the data packets, suppose that the source transmits an additional stream of low-priority packets each of which are independent, random linear combinations of the data packets transmitted over the past d units of time. In other words, each low-priority packet is simply a random weighted sum of all the data packets that were transmitted over the past d units of time. At each of the intermediate nodes in the network, during congestion (i.e., the number of data packets plus the number of coded packets exceeds the link capacity), some of these coded packets are preferentially dropped. In other words, nodes in the network employ a two-level priority scheduling, where data packets are transmitted with higher priority than coded packets. Note that if the total

[1]We use Knuth's notation $O(n)$, $\Theta(n)$, $\Omega(n)$ to denote functions that scale slower than (upper bounded by), as fast as (upper and lower bounded by positive constants) and faster than (lower bounded by) n respectively.

number of data packets arriving in a time-slot exceeds the link capacity, some data packets will be dropped as well. The decoder at the receiver can then recover the *lost data packets* if it receives a suitable number of *random linear coded packets* within an interval of time of d units.

We illustrate this in the context of a path in a network (see Figure 4.1), where a data flow passes through a sequence of nodes in the presence of cross traffic. In a conventional buffered network, each intermediate node needs packet buffers to temporarily store packets when bursts of data packets arrive. On the other hand, in the network coded case (with zero buffers at intermediate nodes), a coding buffer at the source needs to maintain a window of packets (over a time-interval of d units).

Spatial buffer multiplexing can result in significant gains in buffer requirements. Consider, for example, for a rectangular grid network with N nodes which are randomly partitioned into $N/2$ sources matched to $N/2$ destinations. The typical path contains $\Theta(\sqrt{N})$ links and each link carries on an average $\Theta(\sqrt{N})$ flows through it. With a buffer of size b for each flow at each intermediate router, the total number of buffers per-flow is $\Theta(\sqrt{N}b)$. Now, since there are $N/2$ flows, the total number of buffers required across the network is $\Theta(N\sqrt{N}b)$. In contrast, we will show that using network coding with RLCs of $d = \Theta(b)$ time-steps, each source-destination pair requires a (coding) buffer of size $\Theta(b)$ only and no buffers are required at the intermediate nodes. Hence, the total number of buffers required across the network is $\Theta(Nb)$. This comes as an average $\Theta(\sqrt{N})$ buffer-size gain over traditional queueing.

In this chapter, we first consider a large network with many nodes and many flows through each link(edge) in the network to compare alternate strategies. We employ many-sources large deviations analysis [55],[56] to quantitatively demonstrate that the packet loss probabilities with these two strategies are orderwise similar in the exponent. Large deviations have been used to analyze packet-loss, delay and other QoS parameters in networks with large number of sources (many sources large deviations)[56],[51],[50] or with large buffers (large buffer large deviations)[57]. In the context of many sources large deviations, a rate function indicates that the probability that a QoS parameter is not met decreases *uniformly* in the exponent with the number of sources. Botvich and Duffield [50] show that the queue length Q^n at the head of a link exceeds the buffer size nb is given by the rate

$$\lim_{n\to\infty} \frac{1}{n} \log P(Q^n > nb) = -I(b). \tag{4.1}$$

Further, the authors show that for uncorrelated arrivals at the queue, $I(b) \approx \delta b + \nu$ for some $\delta > 0$, i.e. the rate function $I(b)$ is linear in b in the large b regime.

4.1.2 Small networks: Large source buffer

In the discussion so far, we have mainly focussed on the analysis of the large network case and the performance of spatial buffer multiplexing in the case of large networks. However, from the perspective of overall system design, it is important to study the performance of RLC at the source even when it is not possible to have stochastic multiplexing with many other sources at a node. To study the effect on delay and packet loss in the case of a single bursty source-destination pair and compare queueing and coding in this regime, we study the large-buffer packet-drop probability (QoS) of coding and compare that against queueing. Thus, we need to determine the conditions such that we achieve the same QoS requirements if we used coding at the source instead of using a network buffer at the point of entry in the network.

We will subsequently demonstrate that in this case, traditional queueing performs much better than coding at the sources.

4.1.3 Main Contributions

We consider the comparison of buffering at each intermediate link along a path versus network coding at the source and decoding at the destination. We first consider the case of a single *bufferless* link with capacity nC packets per time-slot where n is the number of flows, with mean arrival rate $E[A^m]$ for the m-th flow, through this link, and $C > E[A^m]$, for all $m = 1, 2, \ldots n$, is the capacity per-flow for this edge. We assume that RLCs of packets in d previous time-slots are transmitted as a lower-priority auxiliary coded packet stream. In this context, we obtain the many sources large deviations rate function for packet loss across this edge as an increasing function of d. Subsequently, we generalize this result to the case of a path where the number of edges(links) is a polynomial in n_e, the number of flows through each edge e in the path.

A preliminary overview of large deviations is presented in Section 4.2. Section 4.3 presents a detailed system model for the encoder and decoder, a quick overview of Random Linear Coding and describes the *proportional dropping* rule where, in the event of overflow, packets are dropped from flows in proportion to the size of each flow. We also state the conditions under which packets dropped in previous time-slots can be recovered with the aid of coded packets in subsequent time-slots.

Our main contributions are as follows:

(i) Since RLC couples the packet drop in one time-slot with the arrival rates in the past and future time-slots, deriving the exact expression for the probability of packet loss is difficult. In Section 4.4 we upper bound the probability of packet loss over a link with n flows through it by $\exp(-nI_Y(0,d,\bar{B}))$ where $I_Y(0,d,\bar{B}) > 0$ is a increasing function in d. We further derive a lower bound to show that the above bound is orderwise tight in the exponent. Further, in Section 4.7 we show that for i.i.d. Bernoulli arrivals, $I_Y(0,d,\bar{B}) = dK_1$ for some constant $K_1 > 0$. This implies that the probability of a packet loss decreases exponentially with n and d which compares with the many sources queueing result of Botvich and Duffield [50], Equation (4.1). We plot the packet loss probabilities with network coding in comparison with buffering and show that if the buffer required for coding is orderwise the same as the buffer for queueing, the same QoS (packet loss probability) can be obtained.

(ii) In Section 4.5, we generalize the rate function to the case of a path with multiple links and for coding buffer of $d = \Theta(1)$. We derive an upper bound on the probability of packet drop that decays exponentially in n_Γ, the minimum number of flows through any edge along path Γ. We numerically show in Section 4.7 that the rate function is asymptotically linear in d.

In large networks with N nodes where $n_\Gamma = \Omega(N^\alpha)$, $\alpha \in (0,1)$,(see Section 4.5 for networks with this property) we argue that for achieving comparable QoS, `(buffer per node with traditional queueing)/(buffer per node with network coding)` $= \Omega(N^\alpha)$. This order-wise buffer savings makes a case for the use of network coding for *spatial buffer multiplexing* in favour of queueing at intermediate routers for such networks.

(iii) Theorem 7 in Section 4.6, presents a sufficient condition for the packet loss probability to decline exponentially in the size of the delay (and in turn linearly to the RLC source-coding buffer. Further, we present a representation of the sufficiency rule in terms of a *loss effective bandwidth* and observe that our sufficient condition for the loss effective bandwidth under coding is similar to the necessary and sufficient condition for the 'buffer effective bandwidth' (under queueing) as described by de Veciana and Walrand [57].

Finally, we present numerical results comparing the loss effective bandwidth for the queueing against coding and state a few properties of the delay effective bandwidth in Lemma 21.

As a technical aside, we note that network-wide many-sources or large-buffer large deviations analysis with traditional buffering at intermediate nodes is very difficult due to the correlation of processes in links along a path. However, network coding allows sufficient decoupling that enables our analysis in the network-wide context.

4.2 Preliminaries and Prior Work

For a large network with many source-destination pairs, under fairly general topology assumptions, each link carries the load of multiple source-destination pairs. Assuming that the link capacities scale orderwise linearly with the number of flows through a link, so as to allow $\Theta(1)$ per-flow capacity at each link, we can quantify various QoS properties of the flows, such as packet drop probability and maximum delay, in terms of large deviations rate functions of the arrival and service processes at the link queues [50],[57],[56],[51].

4.2.1 Large deviations

For a sequence of i.i.d. random variables $X_1, X_2, \ldots$ where $E[X_i] = \hat{X}$, the Strong Law of Large Numbers states that the empirical mean $X^{(n)} = \frac{1}{n}\sum_{i=1}^{n} X_i \rightarrow \hat{X}$ *almost surely* in the limit as $n \rightarrow \infty$. In the pre-limit, for finite n, Chernoff's bound characterizes the rate of convergence of $X^{(n)}$ to the mean $\hat{X}$ as follows,

$$P(|X^{(n)} - \hat{X}| > \delta) \leq 2\exp\left[-n \sup_{\theta}(\delta\theta - \log M_X(\theta))\right]$$

where $M_X(\theta) = E[\exp(\theta(X_1 - \hat{X}))]$ is the log moment generating function of the zero mean process $X_i - \hat{X}$. Further [56][55], it can also be shown that the above bound is tight. Thus, for any $\epsilon > 0$, there exists an n_ϵ such that for all $n > n_\epsilon$,

$$P(|X^{(n)} - \hat{X}| > \delta) \geq 2\exp\left[-n \sup_{\theta}(\delta\theta - \log M_X(\theta) + \epsilon)\right].$$

We can therefore state that the sequence of random variables $X^{(1)}, X^{(2)}, \ldots$, converges to $\hat{X}$ with a *large deviation property* with *rate function*

$$I(x) = \sup_{\theta}\{\theta x - \log M_X(\theta)\}.$$

The rate function $I(x) \geq 0$ since setting $\theta = 0$, $0.x - \log M_x(0) = 0$.

Thus the large deviations rate function gives an understanding of how fast a sequence of random variables converges to the typical value of the sequence as we consider

increasingly large numbers of these variables. This analysis can be extended to the case a general sequence of random variables as follows. A sequence of random variables $Z_1, Z_2, \ldots$ is said to satisfy a large deviations principle with rate function $I_Z(\cdot)$ if for every Borel set A,

$$\begin{aligned} -\inf_{z\in A^0} I_Z(z) &\leq \lim_{n\to\infty} \inf \frac{1}{n} \log P(Z_n \in A) \\ &\leq \lim_{n\to\infty} \sup \frac{1}{n} \log P(Z_n \in A) \leq -\inf_{z\in \bar{A}} I_Z(z) \end{aligned} \tag{4.2}$$

where A^0 and $\bar{A}$ are the interior and closure of set A [56],[55].

In the following sections, we will study the sequence of random variables $f(X^{(1)})$, $f(X^{(2)}), \ldots$, where each $X^{(n)} \doteq \frac{1}{n}\sum_{m=1}^{n} A^m$ is the empirical average of n independent identically distributed (i.i.d.) random variables A^m, $m = 1, 2, \ldots, n$ and $f()$ is a continuous function. Note that in general, A^m can be either a scalar or a vector random variable.

4.2.2 Large buffer large deviations and effective bandwidth

The problem of characterizing the rate at which the probability of losses (packet drops) decays as a function of the buffer size in queueing networks has been examined using 'large-buffer' deviations techniques in [77],[78],[81],[79], [80] and [57].

Consider a stationary ergodic discrete-time random process $\{A_i\}$ where A_i denotes the number of packets arriving at the source in time-slot i with mean $E[A]$. These packets are transmitted (serviced) on a link of rate (capacity) $C > E[A]$ with a buffer provisioning of size b to guard against packet drops due to bursty arrivals.

Chang [78], [77], de Veciana and Walrand [57] present a necessary and sufficient condition for which the asymptotic packet drop probability scales linearly in the exponent with the buffer size b at some rate $\delta > 0$ as follows,

$$\frac{\Lambda_A(\delta)}{\delta} < C \Leftrightarrow \lim_{b\to\infty} \frac{1}{b} \log P(B > b) \leq -\delta \tag{4.3}$$

where $\{B > b\}$ denotes the event that the buffer size random variable B exceeds threshold b, and Λ_A is the cumulant generating function of the arrival process $\{A_i\}$ given by

$$\Lambda_A(\theta) = \lim_{t\to\infty} t^{-1} \log E \exp\left[\sum_{i=1}^{t} A_i\right].$$

Observe that in (4.3), the condition $\Lambda_A(\delta)/\delta < C$ in terms of the arrival process and the decay rate δ is analogous to the stability requirement in queueing that $E[A] < C$.

Accordingly, the expression $\Lambda_A(\delta)/\delta$ is termed the *'effective bandwidth'* of the arrival process as a function of the QoS rate δ. For a more detailed exposition of the literature on effective bandwidth results, including results on the mixture of multi-class stationary ergodic traffic, we refer the reader to the detailed survey contained in the introduction in [57].

4.3 System Models

We will consider a single-source destination pair (single source stream model) under both the finite(Section 4.4) and the large buffer(Section 4.6) regimes. To obtain our results for the large network case, we will need to extend our results in Section 4.4 to the multi-hop multiple source-streams case in Section 4.5. Accordingly, we present two system models – a single source stream model, and its extension to the multiple source stream model.

4.3.1 Single source stream

Consider the simplest case of a single user stream over a single zero buffer link of constant capacity C without delays. We assume slotted time. Also, define the window $W_i \doteq \{t : i - d + 1 \leq t \leq i\}$ of size d corresponding to the i-th time-slot. In time-slot i, the source (head of the link) generates a random number of *data packets* $\{P_{i,j}\}$, $j = 1, 2, \ldots, A_i$ and transmits them across the link. Here we assume that the data packet arrival process $\{A_i\}$, $i = (-\infty, \infty)$ is a *stationary ergodic* random process taking values chosen from a set $\mathcal{A} \subseteq \mathbb{N}$, with mean strictly less than C.

Each packet $P_{i,j}$ can be assumed to be a vector of size s containing elements $P_{i,j}(m)$, $m = 1, 2, \ldots, s$ chosen from a finite field $\mathbb{F}_q$. In general therefore, each $P_{i,j} \in \mathbb{F}_q^s$. The source also generates a low-priority auxiliary data stream of B *coded packets* $\{P'_{i,j}\}$ by an RLC over all packets in the window W_i according to the rule:

$$P'_{i,j}(m) \doteq \sum_{t \in W_i} \sum_{k=1}^{A_t} \alpha_{t,k} P_{t,k}(m) \tag{4.4}$$

for all $j = 1, 2, \ldots, B_i$ and all $m = 1, 2, \ldots, s$ where each $\alpha_{t,k}$ is a random element in $\mathbb{F}_q$ and all arithmetic is over $\mathbb{F}_q$. If $A_i + B_i > C$, priority is given to the data packets $P_{i,j}$ over the coded packets. The purpose of the coded packets is to help recover packets that were lost in any of the past d time-slots. In this sense, the auxiliary data may be thought of being generated by a random linear convolutional encoder with memory d at the source, see Figure 4.2. Note that the link constraint implies that the number of auxiliary data packets

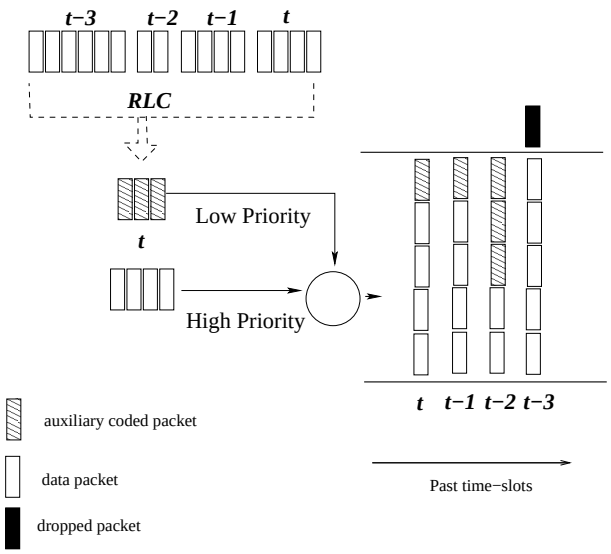

Figure 4.2: Illustration of RLC across d time-slots for a particular source for $d = 4$: each small blank rectangular tile represents a data packet. RLC is performed over all the data packets in the previous $d = 4$ time-slots to generate $\bar{B} = 3$ auxiliary coded packets (shaded tiles) each time-slot. Data packets have higher priority in the link with capacity $C = 5$. The auxiliary coded packets have lower priority and are sent when there is spare capacity in the link. The dark tile represents the dropped packet at time-slot $t - 3$ when 6 packets were generated since the link capacity is only 5.

received by the destination (tail of the link) at time t is $\min(\bar{B}, (C - A_t)_+)$.

Denote the number of lost packets in time-slot i by L_i where $x_+ \doteq \max\{x, 0\}$. For the single source case, $L_i \doteq (A_i - C)_+$. When a packet is dropped, without loss of generality, at time-slot 0, the receiver attempts to recover the dropped packets by decoding the coded packets received in future time-slots by solving for the unknown values of $P_{i,j}$ from the set of equations in (4.4).

The destination receives the the coefficients of the linear equations, $\alpha_{t,k}$, corresponding to each coded packet as header bits within the packet. Alternately, since in most practical considerations, the coefficients $\alpha_{t,k}$ will be generated via a pseudo-random generator, it may be sufficient to initialize the seeds of the pseudo-random generators at the source and destination to the same state at the beginning of the communication process via some form of handshaking. However, this would require the decoder at the receiver to know the exact number of packets generated in each time-slot so as to maintain both random-number generators at the same state. This information could be encapsulated as part of one or more of the data packets.

Each auxiliary coded packet, together with the corresponding coefficients $\alpha_{t,k}$, represents a linear equation over the data packets $P_{i,j}$. As such, the set of regular data packets and auxiliary coded packets at the decoder may be represented as a set of linear equations in *known* and *unknown* variables. The *known* variables correspond to the data packets are directly received by the decoder. The *unknown* variables are the dropped packets.

Hence, the decoder requires as many independent linear equations (coded packets) as the number of unknowns to be able to solve for this set of equations. Note that since the field $\mathbb{F}_q$ is finite, in general, two coded packets have a non-zero probability of being linearly dependent. This corresponds to the event where the matrix of coefficients is singular. In the rest of this work we will loosely refer to the set of linear equations as being *invertible* (uninvertible) if this matrix is not invertible (respectively, not invertible).

Since packets that are dropped can be recovered in a future time-slot, we make a distinction between dropping a packet and losing a packet as follows. L_i packets are said to be *dropped* at time-slot i if $A_i > C$. However, some of these dropped packets may be *recovered* by future coded packets. Hence, packets are said to be *lost* if they are dropped and cannot be recovered by solving for the linear equations formed by the coded packets. Observe that the encoding process couples the loss of a packet in one time-slot with losses

in the past and the future. This cascading effect implies that a packet that is transmitted at time 0, may be decoded in the distant future (possibly after infinite delay) when the set of linear equations is solvable.

However nearly all practical applications require that all packets must be decoded within finite delay. This motivates an additional QoS condition requiring a packet to be decoded within d time-slots. Conversely, a dropped packet that is not decoded within d time-slots is considered *lost* by the decoder at the destination.

4.3.2 Multiple source streams: finite buffer

In general, a link in a large network transmits packets from a large number of sources. For the subsequent analysis we will assume that the link capacity scales in proportion to the number of sources transmitting over the link. The number of sources transmitting over a link depends, in general, on the total number of nodes N, the topology of the network and the number of simultaneous source-destination pairs transmitting. For simplicity, in Section 4.4, we will first deal with the abstraction of a link with n source streams over a single bufferless link of capacity nC packets/time-slot.

Each source S_m, $m = 1, 2, \ldots, n$ generates A_t^m packets $P_{t,j}^m$, $j = 1, 2, \ldots, A_t^m$ in time-slot t. A total of $(\sum_{m=1}^n A_t^m - nC)_+$ packets will be dropped in each slot t. However, the distribution of the dropped packets is a function of the dropping rule at the head of the link. We define the *proportional dropping* rule where

$$L_t^m \doteq \frac{A_t^m}{\sum_{m=1}^n A_t^m} \left(\sum_{m=1}^n A_t^m - nC \right)_+ \tag{4.5}$$

are dropped from the m-th stream at time t. We assume $L_t^m \doteq 0$ for empty links $\sum_{m=1}^n A_t^m = 0$.

If $\sum_{m=1}^n A_t^m < C$, the residual capacity is split equally between coded packets from each source. Thus the number of coded packets from source S_1 received at the tail of the link is

$$B_t^m \doteq \min\left(\bar{B}, \left(C - \frac{1}{n} \sum_{m=1}^n A_t^m \right)_+ \right). \tag{4.6}$$

Subsequently, in Section 4.5, we will extend the path loss results to links in a path Γ of length N^α in a network of size N.

4.4 Probability of packet loss: Many sources

Let $\mathcal{E}_T$ be the event that the last window where no packets were dropped from this stream was W_{-T}. Also, let $D_1^{(n)}$ be the random variable denoting the delay within which all packets $P_{0,k}^1$, $k = 1, 2, \ldots, A_0^1$ are successfully received (directly, or via decoding future coded packets). In keeping with the QoS requirement therefore, packets dropped at time 0 (if they are dropped) will be recovered if and only if $D_1^{(n)} \leq d$, i.e. the decoding delay is less than or equal to d. Due to the interdependence of decodability across time-slots, the exact expression for $P(D_1^{(n)} > d)$ is difficult to compute and so we will attempt to bound this value.

For a finite field $\mathbb{F}_q$, a random matrix has a finite probability of not being invertible.

Condition 1. *If the number of linear equations is greater than or equal to the number of unknowns, the set of linear equations is solvable for the unknowns if the coefficient matrix of the linear equations is invertible.*

We also use $\mathcal{S}_k$ to denote the event that the coefficient matrix corresponding to the RLCs in window W_k is invertible.

For the rest of this chapter, we use the notation $\{\mathcal{C}\}$ to denote the event set $\{\omega : \omega \in \Omega,\ \omega \text{ satisfies condition } \mathcal{C}\}$ where $\Omega = \prod_{m=1}^{n} \Omega_{A^m} \times \Omega_{B^m}$ is the total sample space represented as a product space of the sample path spaces of the packet arrival processes A_t^m and B_t^m. For example $\{D_m^{(n)} > d\} \doteq \{\omega : \omega \in \Omega, D_m^{(n)}(\omega) > d\}$ is the set of sample paths corresponding to the event that the decoding delay for flow from source S_m is greater than d. The complement of an event $\{\mathcal{C}\}$ will be denoted by $\{\neg\mathcal{C}\}$.

Observe that by definition, the sets $\mathcal{E}_T$ are disjoint for different values of T, so if $T \neq T'$, $\mathcal{E}_T \cap \mathcal{E}_{T'} = \emptyset$. Also, since $\mathcal{E}_0$ implies that $L_0^1 = 0$, $P(D_1^{(n)} > d | \mathcal{E}_0) = 0$. Hence we can write

$$P(D_1^{(n)} > d) = \sum_{T=1}^{\infty} P(\{D_1^{(n)} > d\} \cap \mathcal{E}_T). \tag{4.7}$$

4.4.1 Upper bound

To obtain the upper bound, we first find a superset of the set of sample paths corresponding to the event $\{D^{(n)} > d | \mathcal{E}_T\}$ in the following lemma.

Lemma 15. $\{D_1^{(n)} > d\} \cap \mathcal{E}_T \subseteq \left[\bigcup_{k=-T}^{d} \{\{\sum_{i \in W_k} L_i^1 > \sum_{i \in W_k} B_i^1\} \cup \{\neg \mathcal{S}_k\}\}\right] \cap \mathcal{E}_T$.

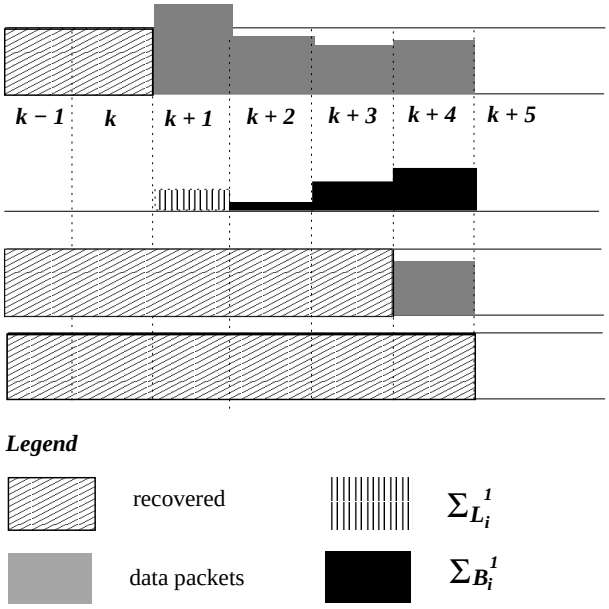

Figure 4.3: Progression of the Induction over each $j^* \geq 1$:

Proof: We proceed by framing the contrapositive[2].

$$\mathcal{E}_T \cap \bigcap_{k=-T}^{d} \{\{\sum_{i \in W_k} L_i^1 \leq \sum_{i \in W_k} B_i^1\} \cap \{\mathcal{S}_k\}\} \quad \subseteq \quad \{D_1^{(n)} \leq d\} \cap \mathcal{E}_T. \tag{4.8}$$

In other words, when $\mathcal{E}_T$ holds, it suffices to show that if for each of the consecutive windows W_{-T} to W_d, the number of losses is less than or equal to the number of coded packets and the RLCs in each window are linearly independent, then packets lost at time-slot 0 can be recovered within d time-slots. We now prove by induction over the sequence of windows $\{W_{-T}, W_{-T+1}, \ldots, W_d\}$. Since $\mathcal{E}_T$ is true, the packets in W_{-T} are all directly received by the destination without requiring any decoding.

Induction Hypothesis: Consider any time-slot $T_0 \geq -T$ such that all packets that were dropped between $-T - d + 1$ and T_0 are recovered by T_0. Then there exists a $1 \leq j^* \leq d$

[2]We use the following contrapositive argument: Given any sets A, B, C, we have $[A \cap C] \subseteq [B \cap C] \iff [\neg B \cap C] \subseteq [\neg A \cap C]$.

such that all packets that are dropped between $-T-d+1$ and T_0+j^* are recovered by T_0+j^*, i.e. within d time-slots.

We first show that this is true for the base case, i.e. for $T_0=-T$. Since $\mathcal{E}_T$ holds, no packets are dropped between $-T-d+1$ and $-T$ and hence all packets dropped between $-T-d+1$ and $-T$ are recovered by $-T$. $\mathcal{E}_T$ also implies that there is a packet lost at time-slot $-T+1$, i.e. $L^1_{-T+1}>0$. We now need to find a j^* such that all packets dropped before $-T+j^*$ are recovered by $-T+j^*$ to prove that the induction hypothesis is true for the base case. Now consider the time-slots in window W_{-T+d+1} from $-T+1$ to $-T+d$. Also since the LHS of (4.9) states that condition $\sum_{i\in W_{-T+d}} L^1_i \leq \sum_{i\in W_{-T+d}} B^1_i$ is true, there must be a time-slot

$$-T+j \doteq \arg\min_{t\in W_{-T+d}} \sum_{i=-T+1}^{-T+t} L^1_i \leq \sum_{i=-T+1}^{-T+t} B^1_i. \tag{4.9}$$

In other words, $-T+j$ indexes the first time-slot after $-T+1$ when the number of auxiliary packets *just overshoots* (i.e. becomes greater than or equal to) the number of lost packets till that time-slot. Since $L^1_{-T+1}>0$ also implies that $B_{-T+1}=0$, it must be that $2\leq j\leq d$. Now, all the auxiliary packets from time-slot $-T+2$ to $-T+j$ are RLCs of data packets generated in the time-slots between $-T-d+3$ to $-T+j$. Since the coefficient of the RLCs are all known at the receiver, each RLC can be considered as a linear equation over the set of known, and unknown symbols, in $\mathbb{F}_q$ corresponding to packets that have not been dropped, and those that have been dropped, respectively. By the definition of j in (4.9), the number of unknown symbols in this set of linear equations $\sum_{i=-T+1}^{-T+t} L^1_i$ is matched or exceeded by the number of simultaneous linear equations $\sum_{i=-T+1}^{-T+t} B^1_i$. The LHS of (4.9) also implies that $\mathcal{S}_{-T+j}$ is true, and therefore that these equations are linearly independent, i.e. Condition 1 holds. Consequently, the receiver can solve this set of simultaneous equations (say, by Gaussian elimination), to decode the unknown symbols (dropped packets). Thus, *all* packets that were dropped before $-T+j$ have been recovered at time-slot $-T+j$ for $1<j\leq d$ demonstrating that the base case holds with $j^*=j$.

In general, assume that the induction hypothesis holds for any arbitrary time-slot $k\geq -T$. This means that all packets from $-T-d+1$ to k are known at the receiver. If there is no loss at time-slot $k+1$, then we set $j^*=1$ to observe that the induction hypothesis still holds. If otherwise, i.e. $L^{k+1}_i>0$ (see Figure 4.3), we consider the window W_{k+d} containing time-slots from $k+1$ to $k+d$. Then, analogous to the base case, we can find a $1<j'\leq d$

such that

$$k + j' \doteq \arg \min_{t \in W_{k+d}} \sum_{i=k+1}^{k+t} L_i^1 \leq \sum_{i=k+1}^{k+t} B_i^1.$$

Again, noting that Condition 1 holds, we have a set of linearly independent simultaneous equations where the number of unknowns is matched or exceeded by the number of equations. Hence, once again, setting $j^* = j'$, we can show that all packets that are dropped between $-T - d + 1$ to $k + j^*$ can be recovered at $k + j^*$.

Since j^* is always greater than 1, the induction proceeds forward along the time-steps where packets are recovered all the way to packets lost in time-slot 0. Also, since $j^* < d$, we can easily see that packets dropped at 0 will be decoded within the next d time-slots.

This proves the contrapositive. We are now done. □

From (4.7) and Lemma 15, for any arbitrarily fixed $\bar{T} > 0$, we conclude using the union bound that

$$\begin{aligned} P(D_1^{(n)} > d) \leq \sum_{T=1}^{\bar{T}} \Bigg(\sum_{k=-T}^{d} P\Big(\sum_{i \in W_k} L_i^1 > \sum_{i \in W_k} B_i^1\Big) \\ + \sum_{k=-T}^{d} P(\neg \mathcal{S}_k) \Bigg) + \sum_{T=\bar{T}+1}^{\infty} P(\mathcal{E}_T). \end{aligned} \tag{4.10}$$

Note that the above expression remains true even as $\bar{T} \to \infty$. We observe that the probability that the matrix of coefficients $\alpha_{t,k}$ will be of non-full rank $P(\neg \mathcal{S}_k)$ depends on the choice of q in $\mathbb{F}_q$. In the sequel, we will first obtain bounds on each $P(\sum_{i \in W_k} L_i^1 > \sum_{i \in W_k} B_i^1)$ and then choose q such that $\sum_{k=-T}^{d} P(\neg \mathcal{S}_k)$ is dominated by $\sum_{k=-T}^{d} P(\sum_{i \in W_k} L_i^1 > \sum_{i \in W_k} B_i^1)$.

Traditional large deviations analysis applied to queueing systems focuses on events concerning the empirical mean of a growing set of random variables. Similarly, in the present problem, we are interested in the strong properties of the empirical mean

$$X_i^{(n)} \doteq \frac{1}{n} \sum_{m=1}^{n} A_i^m \tag{4.11}$$

across source inputs. However, the analysis of the probability of decoding failure is complicated by the fact that the expression for L_i^1 contains both the empirical mean term and the individual value A_i^1 corresponding to the arrivals from the first source. For ease of analysis,

we make the practical assumption of a finite support set for the arrival process below.

Assumption 1. *$\mathcal{A}$ is a finite (bounded) set in $\mathbb{N}$.*

In other words, there is a finite $M \in \mathbb{N}$ such that for all sources S_m and time-slots i, the number of packets from each source is upper bounded, i.e. $A_i^m < M$. This, together with (4.5) and (4.6), implies that

$$P(\sum_{i \in W_k} L_i^1 > \sum_{i \in W_k} B_i^1) \leq P(\sum_{i \in W_k} \frac{\left(X_i^{(n)} - C\right)_+}{X_i^{(n)}} M - \min(\bar{B}, \left(C - X_i^{(n)}\right)_+) > 0).$$

Further, to characterize rare events, we need to establish regularity properties for the packet arrival process A_i^m. Since the arrival processes at different sources are assumed to be independent, the following assumption suffices.

Assumption 2. *Fix any $d \in \mathbb{N}$ and some window W_k of size d. Define the vector $\bar{A}^m = (A_i^m)_{i \in W_k}$. Then for all $m = 1, 2, \ldots, n$, $\bar{\theta} \in \mathbb{R}^d$, the log moment generating function*

$$\Lambda_{\bar{A}^m}(\bar{\theta}) \doteq \log E\left[\exp(\bar{\theta} \cdot \bar{A}^m)\right] < \infty$$

exists and is finite.

Further, we need to impose a condition of symmetry and independence among the various packet sources. This is essential to the large deviations framework within which we shall analyze the loss and packet recovery processes.

Assumption 3. *The arrival processes $\{A_i^m\}$ are identically and independently distributed with respect to each other. In other words, the arrival processes are i.i.d. across flows, not necessarily i.i.d. across time.*

For $n \in \mathbb{N}$, define

$$\Lambda_{\bar{A}_n}(\bar{\theta}) \doteq \frac{1}{n} \log E\left[\exp(\sum_{m=1}^{n} \bar{\theta} \cdot \bar{A}^m)\right]$$

where $\bar{A}_n \doteq \sum_{m=1}^{n} \bar{A}^m$. By Assumptions 2 and 3,

$$\Lambda_{\bar{A}}(\bar{\theta}) \doteq \lim_{n\to\infty} \Lambda_{\bar{A}_n}(\bar{\theta}) = \Lambda_{\bar{A}^1}(\bar{\theta})$$

exists for all windows W_k.

Hence, from the Gartner-Ellis Theorem, for any $\bar{x} \in \mathbb{R}^d$, $\bar{X}^{(n)} \doteq \frac{1}{n}\bar{A}_n$ satisfies a large deviation property (LDP) with good rate function [55],[56],[57] that is a convex dual[3] of $\Lambda_{\bar{A}}$

$$I_{\bar{X}}(\bar{x}) = \sup_{\bar{\theta}} \left(\bar{x} \cdot \bar{\theta} - \Lambda_{\bar{A}}(\bar{\theta})\right). \tag{4.12}$$

Observe that the function $f : \mathbb{R}^d \to \mathbb{R}$ defined as

$$f(\bar{x}) \doteq \sum_{i=1}^{d} \left[\frac{(x_i - C)_+}{x_i} M - \min(\bar{B}, (C - x_i)_+)\right] \tag{4.13}$$

is a continuous function defined on $\mathbb{R}^d$. Figure 4.5, plots f for the case of $d = 1$. Now, using the contraction principle [55],[56], the sequence of random variables,

$$Y_k^{(n)} \doteq \sum_{i \in W_k} \left[\frac{(X_i^{(n)} - C)_+}{X_i^{(n)}} M - \min(\bar{B}, (C - X_i^{(n)})_+)\right]$$

satisfies an LDP with rate function,

$$I_{Y_k}(y, d, \bar{B}) = \inf\{I_{\bar{X}}(\bar{x}) : f(\bar{x}) = y\}. \tag{4.14}$$

where the inf of an empty set is defined in the usual manner as ∞. We include d as an argument to the rate function since the rate function varies with d. Subsequently, in Section 4.7 we will show that $I_{Y_k}(y, d, \bar{B})$ increases linearly in d for arrival processes satisfying Assumption 3.

In addition, to be able to bound the value of $P(\mathcal{E}_T)$, we will require an assumption on the mixing properties of each arrival process.

[3]The convex dual is otherwise known as the Legendre-Frenchel transform.

We will now make assumptions on the mixing properties of the packet arrival proceses.

Definition 12. *ϕ-Mixing [59]: Let $\mathcal{F}_i^j$ denotes the σ-algebra $\{X_m : i \leq m \leq j\}$ for the random process $X_1, X_2, \ldots$. We say that $\{X_m\}$ is ϕ-mixing if $\phi(\nu) \to 0$ where*

$$\phi(\nu) = \sup\{|P(B|A) - P(B)|\} \tag{4.15}$$

for any $A \in \mathcal{F}_1^k$, $B \in \mathcal{F}_{k+\nu}^{\infty}$.

Definition 13. *M-dependent [59]: Define a random process $X_1, X_2, \ldots$ to be $M-$dependent if*

$$P(AB) = P(A)P(B)$$

for any $A \in \mathcal{F}_1^k$, $B \in \mathcal{F}_{k+M}^{\infty}$. In other words, random processes separated by M are independent of each other.

Processes formed by convolutions of independent random variables are M-dependent.

Assumption 4. *Let the arrival processes satisfy either of the following two conditions:*

A. For any $n \in \{1, 2, \ldots\}$, the n-dimensional vector arrival process $(A_i^m)_{m=1}^n$ is ϕ-mixing with $\phi(\nu) = \rho^\nu$ for some $\rho \in [0, 1)$; or

B. For each m the arrival process A_i^m be $M-$dependent for some finite $M \in [1, \infty)$.

We remark that in the assumption above, $\rho = 0$ or $M = 1$ each imply that the arrival process A_i^m is i.i.d. across time.

In the following, we will bound the value of $P(\mathcal{E}_T)$ for either the ϕ-mixing condition (Assumption 4-A) or the $M-$*dependent* condition (Assumption 4-B).

Lemma 16. *If Assumptions 2, 3 and 4-A (ϕ-mixing arrivals) hold, and $E(A_0^m) < C$ for all $m = 1, 2, \ldots, n$, there exists a fixed $\epsilon_1 > 0$ such that for all $T > 0$ and $n > N_{\epsilon_1}$,*

$$P(\mathcal{E}_T) \quad \leq \quad e^{(\log d - n\epsilon_1)\sqrt{T}} + O(\sqrt{T}\rho^{\sqrt{T}}). \tag{4.16}$$

Remark 10. *Note that for any arbitrary T, the above bound does not scale exponentially in n, but in T. However, as we will subsequently show, by choosing T appropriately, this bound is sufficient.*

Proof [Lemma 16]: Define $\mathcal{R}_k$ to be the event that window W_k has no packet drops for packets from source 1 with probability

$$P(\mathcal{R}_k) = P(\{\bigcap_{i\in W_k}(L_i^1 = 0)\}).$$

Therefore,

$$\begin{aligned} P(\neg\mathcal{R}_k) &= P(\{\bigcup_{i\in W_k}(L_i^1 > 0)\}) && (4.17)\\ &\leq \sum_{i\in W_k} P(L_i^1 > 0) && (4.18)\\ &= dP(\sum_{m=1}^{n} A_0^m > nC) && (4.19)\end{aligned}$$

where (4.18) follows from the union bound and (4.19) follows from the ergodicity of the arrival process. Also, since $E(A_0^m) < C$ and Assumption 2 is satisfied, the large deviations bound on $P(\sum_{m=1}^{n} A_0^m > nC)$, together with (4.19) implies that there exists a fixed $\epsilon_1 > 0$, and a corresponding $N_{\epsilon_1} \in \mathbb{N}$ such that for all $n > N_{\epsilon_1}$,

$$P(\neg\mathcal{R}_k) \leq de^{-n\epsilon_1}. \quad (4.20)$$

From the definition of $\mathcal{E}_T$, we have that

$$\mathcal{E}_T = \left[\bigcap_{k=-1}^{-T+1}\{\neg\mathcal{R}_k\}\right] \cap \mathcal{R}_{-T} \subseteq \bigcap_{k=-1}^{-T+1}\{\neg\mathcal{R}_k\} \quad (4.21)$$

Choosing only non-overlapping windows every $\sqrt{(}T)$ time-slots apart, we note that

$$\bigcap_{k=-1}^{-T+1}\{\neg\mathcal{R}_k\} \subseteq \bigcap_{j=1}^{\sqrt{T}-1}\{\neg\mathcal{R}_{-j\sqrt{T}}\}.$$

From the above expression and Assumption 4,

$$\begin{aligned} P(\mathcal{E}_T) &\leq P(\bigcap_{j=1}^{\sqrt{T}-1}\{\neg\mathcal{R}_{-j\sqrt{T}}\})\\ &\leq P(\neg\mathcal{R}_{-\sqrt{T}})\left[P(\bigcap_{j=2}^{\sqrt{T}-1}\{\neg\mathcal{R}_{-j\sqrt{T}}\}) + \phi(\sqrt{T}-d))\right] && (4.22)\\ &< P(\neg\mathcal{R}_{-\sqrt{T}})P(\bigcap_{j=2}^{\sqrt{T}-1}\{\neg\mathcal{R}_{-j\sqrt{T}}\}) + \phi(\sqrt{T}-d).\end{aligned}$$

Proceeding similarly for $j = 2, 3, \ldots \sqrt{T}$,

$$P(\mathcal{E}_T) < \prod_{j=1}^{\sqrt{T}-1} P(\neg\mathcal{R}_{-j\sqrt{T}}) + O(\sqrt{T}\phi(\sqrt{T}-d)).$$

Now, using (4.20), and noting that d is finite,

$$P(\mathcal{E}_T) \quad \leq \quad e^{(\log d - n\epsilon_1)\sqrt{T}} + O(\sqrt{T}\rho^{\sqrt{T}}).$$

□

Lemma 17. *If Assumptions 2, 3 and 4-B ($M-$dependent arrivals) hold, and $E(A_0^m) < C$ for all $m = 1, 2, \ldots, n$, there exists a fixed $\epsilon_1 > 0$, and a corresponding $N_{\epsilon_1} \in \mathbb{N}$, such that for all $T > (M+d+1)^2$ and $n > N_{\epsilon_1}$,*

$$P(\mathcal{E}_T) \quad \leq \quad e^{(\log d - n\epsilon_1)\sqrt{T}}. \tag{4.23}$$

Proof: Note that since $T > (M+d+1)^2$, $\sqrt{(}T) > M+d+1$.

The analysis follows trivially as above by setting $\phi(\sqrt{T}-d) = 0$ in (4.22) to obtain

$$P(\mathcal{E}_T) \quad < \quad \prod_{j=1}^{\sqrt{T}} P(\neg\mathcal{R}_{j\sqrt{T}}) \tag{4.24}$$

$$\leq \quad e^{(\log d - n\epsilon_1)\sqrt{T}} \tag{4.25}$$

when $T > M$. □

Note that $P(\mathcal{S}_k)$ is a function of the size of $\mathbb{F}_q$. Most recent work on network coding [17],[13], [12] assumes that the field size is large enough to be able to approximate that the coefficient matrix at the receiver is completely invertible. For a $k \times k$ matrix with elements taken from $\mathbb{F}_q$, the probability that the matrix will not be invertible is $1 - \prod_{l=1}^{k}(1-q^{-l})$. The size of the matrix to be inverted depends on the congestion at the link. For instance, if there is no congestion in the link – an event with high probability, since $E[A] < C$ and n is large – none of the auxiliary coded packets need to be decoded since there are no packet drops. Hence, the size of the matrix that needs to be inverted is equal to the number of drops in d consecutive time-slots. Trivially, the number of auxiliary packets in d slots is bounded by $\bar{B}d$. Hence, for the purposes of our analysis, it is sufficient to bound the field size from below as follows such that Condition 1 always holds.

Assumption 5. *We consider that the field $\mathbb{F}_q$ is large enough so that*

$$1 - \prod_{l=1}^{\bar{B}d}(1 - q^{-l}) \leq P(\sum_{i \in W_0} \left(L_i^1 - B_i^1\right) > 0).$$

Remark 11. *Assumption 5 is easily satisfied in most practical cases. We note that with $\bar{B}, d = 10$, and $P(\sum_{i \in W_0} \left(L_i^1 - B_i^1\right) > 0)$ of the order of 10^{-6} (or 10^{-8}, respectively), this implies that q must be approximately 20 (30, respectively) bits long.*

We are now ready to state our first result.

Theorem 5. *If the average arrival rate for each of $m = 1, 2, \ldots, n$ sources $E(A_0^m) < C$, and $D^{(n)}$ be the delay within which all dropped packets must be recovered, then under the condition that Assumptions 1-5 are satisfied for field size $\mathbb{F}_q$, for any finite $d > 0$,*

$$\lim_{n \to \infty} \frac{1}{n} \log P(D_1^{(n)} > d) \quad \leq \quad -I_Y(0, d, \bar{B})$$

where

$$I_Y(y, d, \bar{B}) \quad = \quad \inf\{I_{\bar{X}}(\bar{x}) : f(\bar{x}) = y\}. \tag{4.26}$$

for the mapping $f(\cdot)$ defined in (4.13).

Proof: Since the processes A_i^m, $m = 1, 2, \ldots, n$ are ergodic and identically distributed, from (4.14) and the definition of the rate function in (4.2), there exists a finite N_0 such that for all $n > N_0$,

$$P(\sum_{i \in W_k} \{L_i^1 - B_i^1\} > 0) \quad \leq \quad \exp\left(-nI_{Y_k}(0, d, \bar{B})\right)$$

for all $k = -1, -2, \ldots - \infty$.

So, defining $I_Y \doteq I_{Y_k}$ and using the upper bound in (4.10), we have for any value of $\bar{T}$,

$$\begin{aligned} P(D_1^{(n)} > d) \quad &\leq \quad \sum_{T=1}^{\bar{T}} 2(T + d) \exp\left(-nI_{Y_k}(0, d, \bar{B})\right) \\ &\quad + \sum_{T=\bar{T}+1}^{\infty} P(\mathcal{E}_T) \qquad (4.27) \\ &\leq \quad \bar{T}(\bar{T} + 2d + 1) \exp\left(-nI_{Y_k}(0, d, \bar{B})\right) \\ &\quad + \sum_{T=\bar{T}+1}^{\infty} P(\mathcal{E}_T). \qquad (4.28) \end{aligned}$$

The 2 in (4.27) stems from Assumption 5 and the consequent bound $P(\neg \mathcal{S}_k) \leq P(\sum_{i \in W_0} \left(L_i^1 - B_i^1\right) > 0)$.

We will now further bound (4.28).

For the ϕ-mixing assumption, recall from Lemma 16 that there exists a fixed $\epsilon_1 > 0$ such that for all $T > 0$ and $n > N_{\epsilon_1}$,

$$P(\mathcal{E}_T) \overset{(a)}{\leq} e^{(\log d - n\epsilon_1)\sqrt{T}} + O(\sqrt{T}\rho^{\sqrt{T}}).$$

Similarly, for the M-dependent assumption, recall from Lemma 17 that there exists a corresponding $N'_{\epsilon_1} \in \mathbb{N}$, such that for all $T > (M + d + 1)^2$ and $n > N'_{\epsilon_1}$,

$$P(\mathcal{E}_T) \overset{(b)}{\leq} e^{(\log d - n\epsilon_1)\sqrt{T}}.$$

Now fix any $n > \max\{N_0, N_{\epsilon_1}, N'_{\epsilon_1}\}$.

Now set $\bar{T} = \max\{n^2 T_0^2, (M + d + 1)^2\}$. Then, from inequalities (a) and (b), for either case in Assumption 4, there exists some constant $K > 0$ such that

$$\begin{aligned} \sum_{T=\bar{T}+1}^{\infty} P(\mathcal{E}_T) &\leq \sum_{T=\bar{T}+1}^{\infty} e^{(\log d - n\epsilon_1)\sqrt{T}} + \sum_{T=\bar{T}+1}^{\infty} K\sqrt{T}\rho^{\sqrt{T}} \\ &\leq K_1 e^{-nT_0K_2} \end{aligned}$$

for some constant $0 < K_1 < \infty$, some finite $K_2 > 0$. Note that in the above expression, constants K_1, K_2, T_0 are all independent of n.

Thus, for either of these cases (ϕ-mixing or M-dependent), (4.28) can be written as

$$\begin{aligned} P(D_1^{(n)} > d) &\leq n^2 T_0^2 (n^2 T_0^2 + 2d + 1) e^{-nI_{Y_k}(0,d,\bar{B})} \\ &\quad + K_1 e^{-nT_0K_2} \qquad (4.29) \\ &\leq n^2 T_0^2 (n^2 T_0^2 + 2d + 1) e^{-nI_{Y_k}(0,d,\bar{B})} \\ &\quad + K_1 e^{-nI_{Y_k}(0,d,\bar{B})} \qquad (4.30) \end{aligned}$$

where (4.30) follows by choosing a fixed T_0 to satisfy $T_0K_2 > I_{Y_k}(0, d, \bar{B})$.

Now, taking the natural logarithm on both sides and dividing by the fixed n,

$$\begin{aligned} &\frac{1}{n} \log P(D_1^{(n)} > d) \\ &\leq 2 \max\left\{ \frac{1}{n} \log(n^2 T_0^2 (n^2 T_0^2 + 2d + 1)), \frac{1}{n} \log(K_1) \right\} - I_{Y_k}(0, d, \bar{B}). \end{aligned}$$

Now, since K_1, T_0 are finite constants independent of n, taking the limit $n \to \infty$, we are done. $\square$

4.4.2 Lower Bound

In this section, we lower bound $P(D_1^{(n)} < d)$ to study the tightness of the upper bound in the previous subsection. We define $\mathcal{E}'_i$ as the event where data packet drops occur in all time-slots in window W_i. Therefore, if $\{L_0 > 0 \cap \mathcal{E}'_d\}$ occurs, then no auxiliary coded packets containing information about the packets lost at time-slot 0 arrive at the destination. Hence, none of the dropped packets can be recovered. Since $\mathcal{E}'_d = \bigcap_{i=1}^{d} \{\sum_{m=1}^{n} A_i^m > nC\}$,

$$P\left(\bigcap_{i=1}^{d}\left\{\sum_{m=1}^{n} A_i^m > nC\right\}\right) \quad \leq \quad P(D^{(n)} > d). \tag{4.31}$$

Assumption 6. *The packet arrival process at each source S_m, $\{A_i^m\}$ is i.i.d. in time, i.e. for two time-slots i, j: $i \neq j$ A_i^m is independent of A_j^m and the two random variables are identically distributed.*

In particular, if Assumption 6 holds, the lower bound in the above expression can be evaluated exactly in terms of the rate function of A_m^1 as follows,

$$P\left(\bigcap_{i=1}^{d}\left\{\sum_{m=1}^{n} A_i^m > nC\right\}\right) \quad = \quad \left[P\left(\sum_{m=1}^{n} A_i^m > nC\right)\right]^d. \tag{4.32}$$

Let

$$\Lambda_A(\theta) \quad \doteq \quad \log E[\exp(\theta A_m^i)] \tag{4.33}$$

be the log moment generating function of the random variable $\{A_m^i\}$, $i = -\infty, \ldots, -1, 0, 1, \ldots, \infty$, $m = 1, 2, \ldots, n$. Note that since the sources have i.i.d. arrival processes, we do not index the expression for log MGF by time-slot i or source m, and will use the same expression for the arrival process from any source at any time.

Since $E[A_i^m] < C$, we have from [55],

$$P\left(\sum_{m=1}^{n} A_i^m > nC\right) \quad \geq \quad \exp(-n\Lambda_A^*(C) + o(n))$$

where

$$\Lambda_A^*(x) \doteq \sup_{\theta}(\theta x - \Lambda_A(\theta)) \tag{4.34}$$

and a function $f(n) = o(n)$ if $\lim_{n\to\infty} f(n)/n = 0$.

Then, from (4.31), (4.32) and (4.34), we arrive at the following result.

Lemma 18. *If each of the sources $m = 1, 2, \ldots, n$ has i.i.d. arrival process $\{A_i^m\}$ (i.e. under Assumption 6), with $E(A_0^m) < C$, and $D^{(n)}$ be the delay within which all dropped packets must be recovered, then for any finite $d > 0$,*

$$\lim_{n\to\infty} \frac{1}{n} \log P(D^{(n)} > d) \quad \geq \quad -d\Lambda_A^*(C)$$

□

We note that the determining lower bound on the limit of $\frac{1}{n}P(D^{(n)} > d)$ as $n \to \infty$ for Markov arrival process at each source in general remains an open problem. Further, we conjecture that the upper and the lower bounds of the limit above are identical in the order of d for the general Markov arrival process case.

4.5 Multi-hop networks: Many sources

In this section, we extend the large deviations results of the previous section from a single link to a general multi-hop network. Recall that we had selected the $\bar{B}$ as the constant rate at which auxiliary data packets are generated by the the source for the single link case. However, in a multi-link path Γ of length $|\Gamma|$ from source N_0 to destination N_L where intermediate nodes $N_1, N_2, \ldots, N_{|\Gamma|-1}$ function as either sources or sinks for their respective streams and well as routing packets destined for other hosts, the rate of auxiliary packets arriving at destination $N_{|\Gamma|}$ is a function of the aggregate traffic flow across all intermediate links. This coupling of the sample paths of each individual source process motivates an approach based on decoupling flows to obtain an appropriate bound on the end-to-end probability that a packet transmitted at time-slot 0 will be *lost*.

We also note that the number of paths n_e crossing a link(edge) e is a function of the topology of the network and the source-destination partition of the nodes in the network. We will assume that at each edge, the capacity of the edge scales as $n_e C$ to ensure that no source-destination paths a completely blocked. For a path Γ defined as a set of edges $e_{N_k,N_{l+1}}$, along the path, we define

$$n_\Gamma \doteq \min_{e\in\Gamma} n_e. \tag{4.35}$$

Assumption 7. *We consider networks where for each edge e in the network $n_e = \Omega(N^\alpha)$ where N is the number of nodes in the network uniformly for some fixed $\alpha \in (0, 1)$. Also the path length $|\Gamma| = O(N^\beta)$ for some $\beta \in (0, 1)$.*

This assumption is motivated by the spate of recent results in scaling laws over large networks [45],[49],[48],[47] such as ad-hoc networks or in server grids. The authors in [48] prove that if N nodes are scattered uniformly over a unit area, divided into sure tiles of area $a(N)$ each, and under a relaxation of the Protocol Model for wireless ad-hoc networks proposed in Gupta and Kumar [45], the number of paths crossing each tile is $O(N/\sqrt{a(N)})$ with high probability when the propagation occurs along a straight line path. Further, for direction based routing with errors but with a *progressive routing* assumption where the distance between the source and destination is reduced by at least $\delta\sqrt{a(N)}$ for some $\delta > 0$ and $a(N) = \frac{\log N}{N}$, Subramanian and Shakkottai [47] show that the total number of tiles $|\Gamma|$, that a path can touch is upper bounded by $\frac{1}{\delta K a(N)}$ for some $K \in \Theta(1)$. Thus, since the mean Euclidean path length is $\Theta(1)$, by symmetry the probability that a path crosses a given tile is lower bounded by $\delta K a(N)$.

For a symmetric rectangular grid of N computers, ignoring edge effects (or assuming a wrap-around at the edges to form a torus) and source-destination pairs chosen uniformly at random from among the nodes, the expected number of paths through any edge is $\sqrt{N}$. This, again points to the validity of Assumption 7.

Assumption 7 together with the definition in (4.35) implies that $n_\Gamma = \Omega(N^\alpha)$ for any path Γ in the network.

Further, by Assumption 5, we will consider the field size (packet size) is large enough such that a lost packet can be decoded simply if the number of auxiliary packets is greater than the number of lost packets in window.

Let $A_{i,e}^m$ be the flow from source m through edge e at time i. Then we define $X_{i,e}^{(n)} \doteq \frac{1}{n}\sum_{m=1}^{n} A_{i,e}^m$ as the normalized cumulative flow of data packets through e at time i.

Further, we use $L_{\Gamma,i}^m$ to denote the number of packets from source S^m *dropped* in time-slot i along path P. Recall that we assume that there are no packet transmissions delays and that we treat each link as a pipe that instantaneously transfers the packet from source to destination in case there is sufficient capacity, else the packet is dropped at the first edge where there is a congestion. In general, link propagation delays can be handled easily by the appropriate indexing of time at each link along the propagation path. However, we skip the details since it does not affect our analysis in any way.

Also, let $B_{\Gamma,i}^m$ be the number of auxiliary packets from source S_m that reach the destination at the end of path P in time-slot i. Also fix any $\bar{T} > 0$. Assuming that the field size $\mathbb{F}_q$ is large enough as before we can bound the term $P(\mathcal{S}_k)$ corresponding to decoding

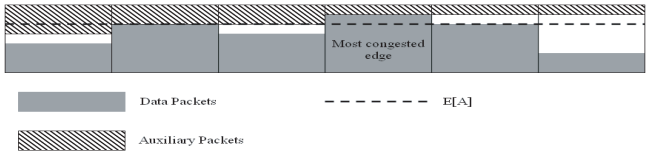

Figure 4.4: The rate of auxiliary packets received at the destination of path Γ is equal to the rate at the tail of the most congested link along P as shown here.

failure in (4.10) (using Assumption 5) to write the probability that packet loss of a packet from source S_m dropped in time-slot 0 on path Γ as

$$\begin{aligned} P(D_{\Gamma,m}^{(n_\Gamma)} > d) \quad &\leq \quad \sum_{T=1}^{\bar{T}} \left(\sum_{k=-T}^{d} 2P\Big(\sum_{i \in W_k} L_{\Gamma,i}^m - B_{\Gamma,i}^m > 0\Big) \right) \\ &\quad + \sum_{T=\bar{T}+1}^{\infty} P(\mathcal{E}_T^\Gamma), \end{aligned} \tag{4.36}$$

where E_T^Γ, analogously, corresponds to the event that the last window where there was no overflow in any of the edges $e \in \Gamma$ was W_{-T}.

Observe that the path packet drop term $L_{\Gamma,i}^m$ is a sum of the edge losses at each edge. However, the edge losses are not independent a each link. Therefore, we bound $L_{\Gamma,i}^m$ by

$$L_{\Gamma,i}^m \leq \bar{L}_{\Gamma,i}^m \doteq M \max_{e \in \Gamma} I_{\{X_{i,e}^{(n_\Gamma)} > C\}} \tag{4.37}$$

where $I_{\{\mathcal{A}\}}$ is the identity function for event $\{\mathcal{A}\}$. The intuition behind the above bound is simple – if the most congested link e along path Γ has $X_{i,e}^{(n_\Gamma)} > C$, then $\bar{L}_{\Gamma,i}^m$ corresponds to the case where the *entire* set of data packets from S_m, which is bounded by M following Assumption 1, is dropped along path Γ.

Note that using the bound in (4.37), we can write the following inequality

$$P\Big(\sum_{i \in W_k} L_{\Gamma,i}^m - B_{\Gamma,i}^m > 0\Big) \leq P\Big(\sum_{i \in W_k} \bar{L}_{\Gamma,i}^m - B_{\Gamma,i}^m > 0\Big). \tag{4.38}$$

Now, assuming that the source generates auxiliary coded packets at a maximum data rate of $\bar{B}$, and using the model of a data-pipe along with packets can be dropped, the rate at which auxiliary packets can reach the destination is determined by the (normalized) cumulative packet $X_{i,e}^{(n_\Gamma)}$ on the most congested link $e \in \Gamma$, see Figure 4.4. Thus,

$$B_{\Gamma,i}^m = \min\left(\bar{B},\ \min_{e \in \Gamma}\ (C - X_{i,e}^{(n_\Gamma)})_+\right). \tag{4.39}$$

Next, it follows that

$$\bar{L}^m_{\Gamma,i} - B^m_{\Gamma,i} = \max_{e\in\Gamma}\left(MI_{\{X^{(n_\Gamma)}_{i,e}>C\}} - \min(\bar{B},(C-X^{(n_\Gamma)}_{i,e})_+)\right).$$

We show this by considering the following two cases. Case (i) occurs when there is no overflow in any link on the entire path, i.e., $\bar{L}^m_{\Gamma,i}=0$, and thus LHS in the equation above is $-B^m_{\Gamma,i}$. The RHS of the equation above, in this case, is

$$\begin{aligned}&\max_{e\in\Gamma}\left(-\min(\bar{B},(C-X^{(n_\Gamma)}_{i,e})_+)\right)\\ &=-\min\left(\bar{B},\min_{e\in\Gamma}(C-X^{(n_\Gamma)}_{i,e})_+\right)=-B^m_{\Gamma,i}=LHS,\end{aligned}$$

and we are done. On the other hand in Case (ii), there is a loss on one (or more) link $e\in\Gamma$ (i.e., $X^{(n_\Gamma)}_{i,e}>C$). In this case, $(C-X^{(n_\Gamma)}_{i,e})_+=0$ and hence $B^m_{\Gamma,i}=0$. Then, we have

$$LHS \;=\; \bar{L}^m_{\Gamma,i} \;=\; M \;=\; RHS,$$

for Case (ii) as well.

This implies that

$$\begin{aligned}&P(\sum_{i\in W_k}\bar{L}^m_{\Gamma,i}-B^m_{\Gamma,i}>0)=\\ &P(\sum_{i=1}^{d}\max_{e\in\Gamma}\left(MI_{\{X^{(n_\Gamma)}_{i,e}>C\}}-\min(\bar{B},(C-X^{(n_\Gamma)}_{i,e})_+)\right)>0).\end{aligned}$$

Let $\mathfrak{L}=\{(e_1,e_2,\ldots,e_d)\}$, $e_i\in\Gamma$, $i=1,2,\ldots,d$. Note that $|\mathfrak{L}|=|\Gamma|^d$. Then we have that

$$\begin{aligned}&\{\sum_{i=1}^{d}\max_{e\in\Gamma}\left(MI_{\{X^{(n_\Gamma)}_{i,e}>C\}}-\min(\bar{B},(C-X^{(n_\Gamma)}_{i,e})_+)\right)>0\}\\ &=\bigcup_{(e_1\ldots e_d)\in\mathfrak{L}}(\sum_{i=1}^{d}\left(MI_{\{X^{(n_\Gamma)}_{i,e_i}>C\}}-\min(\bar{B},(C-X^{(n_\Gamma)}_{i,e_i})_+)\right)>0).\end{aligned}$$

To see this, consider any four random variables Z_1,Z_2,Z_3,Z_4. Then, observe that the event $\{\max(Z_1,Z_2)+\max(Z_3,Z_4)>0\}$ is the same as $\{(Z_1+Z_3>0)\cup(Z_1+Z_4>0)\cup(Z_2+Z_3>0)\cup(Z_2+Z_4>0)\}$ The above statement is merely an extension of this result.

Therefore, using the union bound,

$$\begin{aligned}
&P(\sum_{i=1}^{d}\max_{e\in\Gamma}\left(MI_{\{X_{i,e}^{(n_\Gamma)}>C\}}-\min(\bar{B},(C-X_{i,e}^{(n_\Gamma)})_+)\right)>0)\\
&\leq\sum_{(e_1,\ldots,e_d)\in\mathcal{L}}P(\sum_{i=1}^{d}\left(MI_{\{X_{i,e_i}^{(n_\Gamma)}>C\}}-\min(\bar{B},(C-X_{i,e_i}^{(n_\Gamma)})_+)\right)>0).
\end{aligned}\tag{4.40}$$

Also, since packets can only be dropped from the flow originating from source S_m in subsequent links on the network, we have that any flow $A_{t,e}^m < A_t^m$, where A_t^m is defined as in the previous section to be the total number of data packets generated by S_m in time t. This means that fewer packets are dropped in link e as the link gets farther away from the source S_m since the flow has already been 'thinned out' by dropping packets in the previous links. Hence, we have

$$\begin{aligned}
&\sum_{\mathcal{L}}P(\sum_{i=1}^{d}\left(MI_{\{X_{i,e_i}^{(n_\Gamma)}>C\}}-\min(\bar{B},(C-X_{i,e_i}^{(n_\Gamma)})_+)\right)>0)\\
&\leq\sum_{\mathcal{L}}P(\sum_{i=1}^{d}\Big(MI_{\{X_{i}^{(n_\Gamma)}>C\}}-\min(\bar{B},(C-X_{i}^{(n_\Gamma)})_+)\Big)>0),
\end{aligned}$$

where $X_i^{(n_\Gamma)}$ is as defined in (4.11). Hence, we have from (4.40),

$$\begin{aligned}
&P(\sum_{i=1}^{d}\max_{e\in\Gamma}\left(MI_{\{X_{i,e_i}^{(n_\Gamma)}>C\}}-\min(\bar{B},(C-X_{i,e}^{(n_\Gamma)})_+)\right)>0)\\
&\leq|\Gamma|^dP(\sum_{i=1}^{d}\left(MI_{\{X_{i}^{(n_\Gamma)}>C\}}-\min(\bar{B},(C-X_{i}^{(n_\Gamma)})_+)\right)>0).
\end{aligned}\tag{4.41}$$

However, note that unlike $f(x)$ in Section 4.4, $g:\mathbb{R}^d\to\mathbb{R}$

$$g(x)\doteq\sum_{i=1}^{d}MI_{\{x_i>C\}}-\min(\bar{B},(C-x_i)_+)$$

is not a continuous function and hence, we cannot apply the contraction principle directly. Therefore, we upper bound $g(x)$ by the function $\bar{g}(x)$ as follows. Fix any small $0<\beta<C-\bar{B}$. Then we define

$$\bar{g}(x)\doteq\sum_{i=1}^{d}\bar{g}_i(x_i)$$

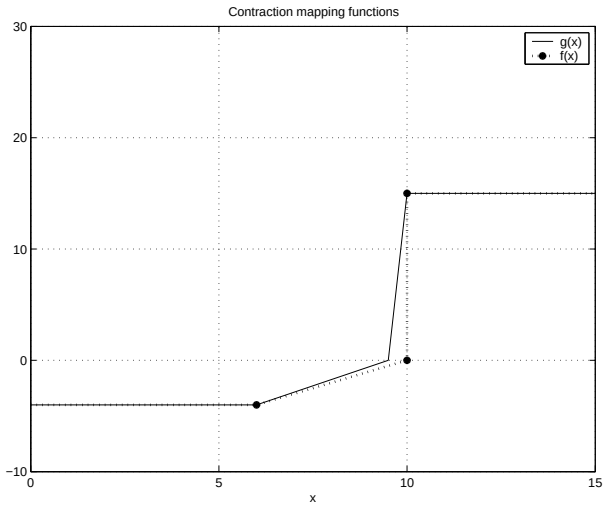

Figure 4.5: Contraction mapping functions f, g and $\bar{g}$ plotted for the case of $M = 15, C = 10, \bar{B} = 3, \beta = 2, E[A] = 8$. Note that the large β is merely for purposes of illustration. A small $\beta > 0$ leads to tighter bounds on the packet loss probability.

where

$$\bar{g}_i(x_i) \doteq \begin{cases} M & \text{for} \quad x \geq C \\ \frac{M}{\beta}(x - C + \beta) & \text{for} \quad C - \beta \leq x < C \\ \frac{\bar{B}}{\bar{B}-\beta}(x - C + \beta) & \text{for} \quad C - \bar{B} \leq x < C - \beta \\ -\bar{B} & \text{for} \quad x < C - \bar{B} \end{cases}$$

as shown in Figure 4.5.

Thus the contraction principle applied to the vector sequence $\bar{X}^{(n_\Gamma)}$ indexed by n_Γ with rate function $I_{\bar{X}}(\bar{x})$ in (4.12), together with (4.41) implies that the sequence of random variables

$$W^{(n_\Gamma)} \doteq \sum_{i=1}^{d} \bar{g}_i(X_i^{(n_\Gamma)})$$

satisfies an LDP with rate function

$$I_W(w, d, \bar{B}) \doteq \inf\{I_{\bar{X}}(\bar{x}) : \bar{g}(\bar{x}) = w\}. \tag{4.42}$$

Thus, we have for n large enough,

$$P(\sum_{i \in W_k} L_{\Gamma,i}^m - B_{\Gamma,i}^m > 0) \quad \leq \quad e^{-n_\Gamma I_W(0,d,\bar{B})}. \tag{4.43}$$

Analogous to Lemma 16 and Lemma 17, we will next prove the exponential tightness of $P(\mathcal{E}_T^\Gamma)$ under the arrival process assumptions of Assumption 4-A or B.

Lemma 19. *If Assumptions 2, 3 and 4-A hold, and $E(A_0^m) < C$ for all $m = 1, 2, \ldots, n_\Gamma$, there exists a fixed $\epsilon_2 > 0$ such that for all $T > 0$ and $n_\Gamma > N_{\epsilon_2}$,*

$$P(\mathcal{E}_T^\Gamma) \leq e^{-n_\Gamma \epsilon_2 \sqrt{T}} + O(\sqrt{T}\rho^{\sqrt{T}}). \tag{4.44}$$

Proof: Define $\mathcal{R}_k^\Gamma$ to be the event that window W_k has no packet drops for packets from source 1 with probability on the entire path Γ,

$$\begin{aligned} P(\mathcal{R}_k^\Gamma) &\triangleq P(\{\bigcap_{e\in\Gamma} \mathcal{R}_k^e\}) \\ &\triangleq P(\{\bigcap_{e\in\Gamma}\bigcap_{i\in W_k} (L_{i,e}^1 = 0)\}), \end{aligned}$$

where

$$L_{i,e}^1 \triangleq \begin{cases} 1 & \text{if} \quad X_{i,e}^{(n_\Gamma)} > C \\ 0 & \text{otherwise.} \end{cases}$$

Also, since $E(A_0^m) < C$ and Assumption 2 is satisfied, there exists a $N_{\epsilon_1} \in \mathbb{N}$ such that for all edges where the number of flows $n_\Gamma > N_{\epsilon_1}$, we can write the analogous relation to (4.20) as follows,

$$P(\neg\mathcal{R}_k^e) \leq de^{-n_\Gamma \epsilon_1}$$

for some fixed $\epsilon_1 > 0$ and any edge $e \in \Gamma$.

Thus,

$$\begin{aligned} P(\neg\mathcal{R}_k^\Gamma) &= P(\{\bigcup_{e\in\Gamma} \neg\mathcal{R}_k^e\}) \\ &\leq \sum_{e\in\Gamma} P(\neg\mathcal{R}_k^e) \\ &\leq |\Gamma| de^{-n_\Gamma \epsilon_1} \end{aligned} \tag{4.45}$$

Observe that by Assumption 7, for $\alpha, \beta \in (0, 1)$ the path length $|\Gamma| = O(N^\beta)$; also the assumption states that $n_e = \Omega(N^\alpha)$ for all edges $e \in \Gamma$ and thus $n_\Gamma = \Omega(N^\alpha)$. Hence, $|\Gamma| = O(N^{\frac{\beta}{\alpha}})$ is polynomial in n_Γ.

Hence, proceeding along the lines of Lemma 16 where we consider non-overlapping windows $W_{j\sqrt{T}}$ for $j = 1, 2, \ldots, \sqrt{T} - 1$, we can then write,

$$P(\mathcal{E}_T^{\Gamma}) \leq \prod_{j=1}^{\sqrt{T}-1} P(\neg \mathcal{R}_{j\sqrt{T}}^{\Gamma}) + O(\sqrt{T}\phi(\sqrt{T})).$$

Now, using (4.45), and the assumption on $\phi()$ from Assumption 4-A, in the above relation, we arrive at (4.44). □

Similarly, using the bound on $P(\neg \mathcal{R}_k^{\Gamma})$ in (4.45), we have the following result for the case where the data arrival process is $M-$*dependent* for $M > (T + d + 1)^2$.

Lemma 20. *If Assumptions 2, 3 and 4-B hold, and $E(A_0^m) < C$ for all $m = 1, 2, \ldots, n_{\Gamma}$, there exists a fixed $\epsilon_2 > 0$ such that for all $T > (M + d + 1)^2$ and $n_{\Gamma} > N_{\epsilon_2}$,*

$$P(\mathcal{E}_T^{\Gamma}) \quad \leq \quad e^{-n_{\Gamma}\epsilon_2\sqrt{T}}. \tag{4.46}$$

□

Substituting (4.43) in (4.36), using the exponential tightness of $P(\mathcal{E}_T^{\Gamma})$ from Lemma 19 (or Lemma 20), choosing a fixed $\bar{T}$ large enough such that the first term in (4.36) dominates (the argument is identical to that in Theorem 5), and noting once again that from Assumption 7, $|\Gamma|^d$ is polynomial in n_{Γ}, we have the probability that a packet dropped on path Γ between source S^m and the destination at time-slot 0 is lost (cannot be recovered) is asymptotically bounded as follows

$$\lim_{n_{\Gamma}\to\infty} \frac{1}{n_{\Gamma}} \log P(D_{\Gamma}^{(n_{\Gamma})} > d) \leq -I_W(0, d, \bar{B}) \tag{4.47}$$

proving the following result.

Theorem 6. *[58] Consider a path Γ from source $S^m = N_0$ to destination $N_{|\Gamma|}$ in a network satisfying the topological requirements in Assumption 7. Also, assume that all sources S^j in the network have packet arrival process, $\{A_i^j\}$ satisfying Assumption 4 with mean $E(A_0^j) < C$. Also, if the source generates auxiliary packets with rate $\bar{B}$, then the probability that a packet dropped from source S^m in time-slot 0 cannot be recovered with delay $D_{\Gamma,m}^{(n_{\Gamma})} < d$ is asymptotically bounded as*

$$\lim_{N\to\infty} \frac{1}{n_{\Gamma}} \log P(D_{\Gamma,m}^{(n_{\Gamma})} > d) \leq -I_W(0, d, \bar{B}) \tag{4.48}$$

□

In the following section, we perform numerical simulations to show that $I_W(0, d, \bar{B})$ is strictly positive and scales linearly in d for i.i.d. arrival processes.

Remark 12. *In a queueing network with buffering at intermediate nodes, each node needs to have a buffer of size $b = \Theta(d)$ allocated for* every *flow passing through it. This follows from many-sources large deviations for a single server queue [50]. Botvich and Duffield show that at a single link, a buffer of $\Theta(n_\Gamma b)$ is necessary to achieve a loss probability that decays as $e^{-n_\Gamma I(b)}$, and $I(b) \approx \delta b + \nu$ (see (4.1)). Consequently, the buffer size at each intermediate node scales similarly (since loss can occur at any of the links in the path of the flow).*

Since, we have assumed that $n_\Gamma = \Omega(N^\alpha)$ (recall from (4.35) that n_Γ is a lower bound on the number of flows through any intermediate router), the above argument implies that the total buffering required in the network (with N nodes) scales as $\Omega(N^{1+\alpha})$.

On the other-hand, for comparable QoS with network coding, Theorem 6 requires $\Theta(d)$ buffers per source-destination flow. *This implies that the total buffer in the network scales as $\Theta(Nd)$ (as there are $\Theta(N)$ source-destination pairs). This gives the* spatial buffer multiplexing *a per-node buffering gain of $\Omega(N^\alpha)$ over traditional queueing at intermediate nodes.*

4.6 Single source large buffer asymptotics

Consider a single-source destination pair under the model of Section 4.3-A. Let d be the maximum number of timeslots over which received packets are buffered at the destination. Here we will consider the packet loss behaviour asymptotically in the large d-regime. Although dropped packets may be recovered infinitely into the future, we impose a QoS requirement where we consider a packet to be lost if cannot be recovered within d timeslots. However, since we are only considering a sufficient condition, this assumption does not impede our analysis. In the rest of this chapter therefore, we will use $P(D > d)$ as a surrogate for the loss probability.

Due to the interdependence of decodability across time-slots, the exact expression for $P(D > d)$ is difficult to compute and so we will attempt to bound this value.

Further, for technical reasons we need to make the following finite history assumption of the system under consideration.

Assumption 8. *Fix any integer $T_0 > 0$. Then, given some coding buffer size d, the system is initialized at $-\bar{T} = dT_0$.*

Note that $T_0 = \infty$, corresponds to the case where the system is observed in steady state. However, for the purpose of this work, we consider the large buffer regime $d \to \infty$ where the coding buffer at time $-\bar{T} = dT_0$ is initialized to zero.

Note that since the events $\mathcal{E}_T$ are disjoint for all T, by the principle of total probability,

$$P(D > d) = \sum_{T=1}^{\bar{T}} P(\{D > d\} \cap \mathcal{E}_T). \tag{4.49}$$

Then from Lemma 15 and (4.49),

$$\begin{aligned} P(D > d) &\leq \sum_{T=1}^{\bar{T}} \left(\sum_{k=-T}^{d} P\Big(\sum_{i \in W_k^{(d)}} (L_i - B_i) > 0 \Big) \right) \\ &= \sum_{T=1}^{\bar{T}} (T + d + 1) P\Big(\sum_{i \in W_k^{(d)}} (L_i - B_i) > 0 \Big) \end{aligned}$$

Since, the link capacity is fixed at C, let $X_i \stackrel{\Delta}{=} A_i - C$ be the effective arriving workload at each time-slot i. By our assumption of stable arrival process, therefore $E(X_i) < 0$. Further, assume that $\{X_i\}$ satisfies a large deviations principle with rate function $I(x) = \sup_\theta(\theta x - \Lambda_X(\theta))$ [4] such that for all $\theta \in \mathbb{R}$ the limiting cumulant generating function

$$\Lambda_X(\theta) = \lim_{d \to \infty} \frac{1}{d} \log E \exp \left[\theta \sum_{i=1}^{d} X_i \right] < \infty, \tag{4.50}$$

and $I(\cdot)$ is strictly convex or that $\Lambda_X(\theta)$ satisfies the requirements of the Gartner-Ellis theorem.

Observe that under the assumption that $\bar{B} = C$, i.e. the maximum number of auxiliary packets generated by the source is C per time-slot, at any time-slot t,

$$L_t - B_t = (A_t - C)_+ - (C - A_t)_+ = A_t - C = X_t.$$

[4] $I(\cdot)$ is said to be the Legendre-Fenchel- or convex- transform of $\Lambda(\cdot)$, see [55] for details.

Then, we can write

$$P(D > d) \quad \leq \quad \frac{1}{2}\bar{T}(\bar{T}+d+1)P(\sum_{t\in W_k^{(d)}} X_t > 0), \tag{4.51}$$

where (4.51) follows from the ergodic nature of the arrival process.

From the Chernoff bound, for any $\theta > 0$

$$P(\sum_{i\in W_k^{(d)}} X_i > 0) \leq E\left[\exp(\theta \sum_{i\in W_k^{(d)}} X_i)\right].$$

Then, from the limit in (4.50) for any $\epsilon > 0$, there exists a d_ϵ such that for all $d > d_\epsilon$

$$E\left[\exp(\theta \sum_{i\in W_k^{(d)}} X_i)\right] \leq \exp[d(\Lambda_X(\theta)+\epsilon)].$$

We can now rewrite (4.51) as follows,

$$P(D > d) \quad \leq \quad \frac{1}{2}\bar{T}(\bar{T}+d+1)\exp[d(\Lambda_X(\theta)+\epsilon)].$$

Further, since from Assumption 8, $\bar{T} = dT_0$ for some constant $T_0 > 0$ and ϵ can be arbitrarily small,

$$\Lambda_X(\theta) < -\delta \text{ implies that } \lim_{d\to\infty}\sup\frac{1}{d}\log P(D > d) \leq -\delta$$

for any $\delta > 0$ and $\theta > 0$. In particular, it suffices that $\inf_{\theta>0}\Lambda_X(\theta) < -\delta$. Since $E[X] < 0$, this is equivalent to stating that $\inf_\theta \Lambda_X(\theta) < -\delta$.

Thus noting that $\inf_\theta \Lambda_X(\theta) = -I(0)$,

$$I_X(0) \geq \delta \implies \lim_{d\to\infty}\sup\frac{1}{d}\log P(D > d) \leq -\delta. \tag{4.52}$$

Theorem 7. *For the system defined in Section 4.3, if the arrival rate $E(A) < C$, and the log m.g.f. $\Lambda_X(\theta)$ for the process $X_i = A_i - C$ is well defined and strictly convex and Assumption 5 holds, then*

$$I_X(0) > \delta \implies \lim_{d\to\infty}\sup\frac{1}{d}\log P(D > d) \leq -\delta. \tag{4.53}$$

□

Now, assume that an aggregate of N flows are created by the source. Further, let there be n_j flows of type $j \in J$ each satisfying a large deviations principle with rate function $I_j()$, and only one flow of coded packets generated by considering an RLC of all packets within a window of d time-slots.

Then, using the contraction principle [55], and (4.53),

$$\inf_{\{\alpha_j : \sum_{j\in J} n_j \alpha_j = 0\}} \sum_{j\in J} n_j I_j(\alpha_j) \geq \delta$$
$$\implies \quad \lim_{d\to\infty} \frac{1}{d} \log P(D > d) \leq -\delta.$$

4.6.1 Loss effective bandwidth representation

In this section we express the necessary condition for QoS in Theorem 7 in terms of an *effective bandwidth* criterion.

First, we will write the condition in (4.52) in terms of the Legendre-Fenchel transform $I_A(x) = \sup_\theta\{\theta C - \Lambda_A(\theta)\}$ of the limiting log moment generating function $\Lambda_A(\theta) = \lim_{n\to\infty} \frac{1}{n} \log E \exp\left[\theta \sum_{i=1}^n A_i\right] < \infty$.

Observing that the random variables A_i and X_i are related according to the relation $A_i = X_i + C$ and applying the contraction principle once again, we note that $I_A(a) = \inf_x\{I_X(x) : a = x + C\} = I_X(a - C)$.

Therefore, we can write the condition in Theorem 7 as follows,

$$I_A(C) > \delta \implies \lim_{d\to\infty} \sup \frac{1}{d} \log P(D > d) \leq -\delta. \tag{4.54}$$

Now let,

$$\inf_{\theta>0}\{(\Lambda_A(\theta) + \delta)/\theta\} < C. \tag{4.55}$$

This implies that for some $\bar{\theta} > 0$,

$$\bar{\theta}C - \Lambda_A(\bar{\theta}) > \delta \implies \sup_{\theta\geq 0}\{\theta C - \Lambda_A(\theta)\} > \delta$$
$$\implies I_A(C) > \delta.$$

Motivated by the above, let us define the loss effective bandwidth

$$C_A(\delta) \stackrel{\Delta}{=} \inf_{\theta>0}\{(\Lambda_A(\theta) + \delta)/\theta\}. \tag{4.56}$$

Note that the loss effective bandwidth for our case corresponds to the probability of the event $\{D > d\}$ and not the buffer overflow probability of [57, 78].

Using (4.55), we can rewrite the result in Theorem 7 as

$$C_A(\delta) < C \implies \lim_{d\to\infty} \sup \frac{1}{d} \log P(D > d) \leq -\delta.$$

The properties of the loss effective bandwidth are listed in the following lemma.

Lemma 21. *(i)* $\lim_{\delta\to 0} C_A(\delta) = E[A]$

(ii) $\lim_{\delta\to\infty} C_A(\delta) = \max_i A_i$

(iii) $C_A(\delta) - \frac{\Lambda(\delta)}{\delta} \leq 1$

Proof: Consider,

$$C_A(\delta) = \inf_{\theta>0} \left[\frac{\Lambda_A(\theta) + \delta}{\theta}\right] \leq \frac{\Lambda_A(K\delta)}{K\delta} + \frac{1}{K}$$

where the last relation follows by substituting any (possibly suboptimal) $\theta = K\delta$ for some $K > 0$. Setting $K = 1$, we see that (iii) follows immediately.

Further, we see that

$$\limsup_{\delta\to 0} C_A(\delta) \leq \lim_{\delta\to 0} \frac{\Lambda_A(K\delta)}{K\delta} + \frac{1}{K} = E[A] + \frac{1}{K} \tag{4.57}$$

since we know from [82] that $\lim_{\delta\to\infty} \Lambda_A(\delta)/\delta = E[A]$.

Also, from Jensen's inequality, $E[e^{\theta \sum_{i=1}^{d} A_i}] \geq e^{d\theta E[A]}$. Since $\theta > 0$ and $\delta \geq 0$,

$$\begin{aligned}
\liminf_{\delta\to 0} C_A(\delta) &\geq \inf_{\theta>0} \left[\frac{\Lambda_A(\theta)}{\theta}\right] \\
&= \inf_{\theta>0} \left[\lim_{d\to\infty} \frac{1}{d\theta} \log E[e^{\theta \sum_{i=1}^{d} A_i}]\right] \\
&\geq \inf_{\theta>0} \left[\lim_{d\to\infty} \frac{1}{d\theta} d\theta E[A]\right] = E[A].
\end{aligned} \tag{4.58}$$

Since we can choose K as large as we want, from (4.57) and (4.58), the result in (i) follows immediately.

Again, since

$$\begin{aligned}
\limsup_{\delta\to\infty} C_A(\delta) &\leq \lim_{\delta\to\infty} \frac{\Lambda_A(K\delta)}{K\delta} + \frac{1}{K} \\
&= \max_i\{A_i\} + \frac{1}{K}
\end{aligned} \tag{4.59}$$

and we can choose $K \in \mathbb{R}$ as large as possible, as $K \to \infty$, we see that $\limsup_{\delta\to\infty} C_\delta \leq \max_i\{A_i\}$ following the known result $\lim_{\delta\to\infty} \Lambda_A(\delta)/\delta = \max_i\{A_i\}$ (cf. [82]).

Now, consider any non-negative sequence $\{\delta_k, k = 1, 2, \ldots, \infty\}$ such that $\delta_k \to \infty$ as $k \to \infty$.

We now show that $\liminf_{k\to\infty} C_A(\delta_k) \geq \max_i\{A_i\}$.

Since the infinite series $\{C_A(\delta_k)\}$ is bounded as above, let $\{\delta_{k_i}, i = 1, 2, \ldots, \infty\}$ correspond to any subsequence (in particular, the subsequence converging to the lim inf) such that $C_A(\delta_{k_i}) \to C^*$ as $i \to \infty$ for some $0 \leq C^* < \infty$. For each i, denote the infimizing $\theta^*(\delta_{k_i}) \in \arg\inf_{\theta>0}\{(\Lambda_A(\theta) + \delta_{k_i})/\theta\}$.

We will consider the two cases:
Case(i): $\liminf_{i\to\infty} \theta^*(\delta_{k_i}) = M < \infty$
Case(ii): $\liminf_{i\to\infty} \theta^*(\delta_{k_i}) = \infty$ and thus $\lim_{i\to\infty} \theta^*(\delta_{k_i}) = \infty$

For case (i), let us denote the subsequence corresponding to the lim inf by $\{\delta_{k_{i_j}}, j = 1, 2, \ldots, \infty\}$, i.e. $\lim_{j\to\infty} \theta^*(\delta_{k_{i_j}}) = M$.

Further, we have by construction that $\delta_{k_{i_j}} \to \infty$ as $j \to \infty$. Thus,

$$C_A(\delta_{k_{i_j}}) = \frac{\Lambda_A(\theta^*(\delta_{k_{i_j}})) + \delta_{k_{i_j}}}{\theta^*(\delta_{k_{i_j}})} \to \infty$$

as $i \to \infty$. However this contradicts the finiteness result in (4.59) and hence case (i) cannot occur.

For case (ii), we have

$$C_A(\delta_{k_i}) = \frac{\Lambda_A(\theta^*(\delta_{k_i})) + \delta_{k_i}}{\theta^*(\delta_{k_i})} \geq \frac{\Lambda_A(\theta^*(\delta_{k_i}))}{\theta^*(\delta_{k_i})} \to \max_i\{A_i\}.$$

Thus, along the sequence $\{\delta_k, k = 1, 2, \ldots, \infty\}$, $\liminf_{k\to\infty} C_A(\delta_k) \geq \max_i\{A_i\}$. Since the sequence $\{\delta_k, k = 1, 2, \ldots, \infty\}$ was chosen arbitrarily, it follows that $\liminf_{\delta\to\infty} C_A(\delta) \geq \max_i\{A_i\}$. Since we have already shown that $\limsup_{\delta\to\infty} C_A(\delta) \leq \max_i\{A_i\}$, the result in (ii) follows immediately.

□

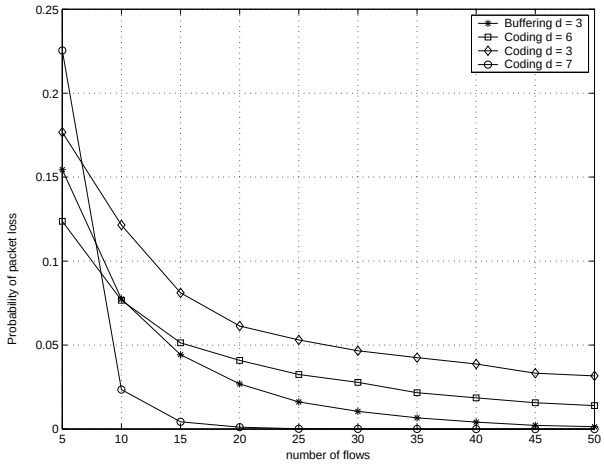

Figure 4.6: Comparison of coding with buffering. Coding with $d = 3, 6$ performs marginally poorer than queueing with $b = 3$. However, coding with $d = 7$ performs better than queueing with $b = 3$. Thus, the performance of coding matches buffering for $d = O(b)$.

4.7 Numerical Results: Many sources

4.7.1 Single Link

Under the i.i.d. Assumption 6, we can show that the rate function for the single link packet loss probability using network coding $I_Y(0, d, \bar{B})$ derived in (4.14) scales linearly in d if the mean arrival rate $E[A] \in (C - \bar{B}, C)$. In this chapter (due to space constraints), we demonstrate this for the simple case where $\{A_i^m\} \sim Bernoulli(p)$ with $p = 0.6$ (hence $E[A_i^m] = 0.6$) over a link of capacity 0.9. The rate function for each A can be derived from the convex dual of the Log Moment generating function (MGF) of the Bernoulli random variable to be

$$I_A(x) = x \log(x/p) + (1 - x) \log((1 - x)/(1 - p)).$$

Using standard rate function computations (for vectors with i.i.d. elements) [55], we can write the rate function for the sequence $\bar{X}^{(n)}$ as

$$I_(\bar{X})(\bar{x}) \quad = \quad \sum_{i=1}^{d} I_A(x_i). \tag{4.60}$$

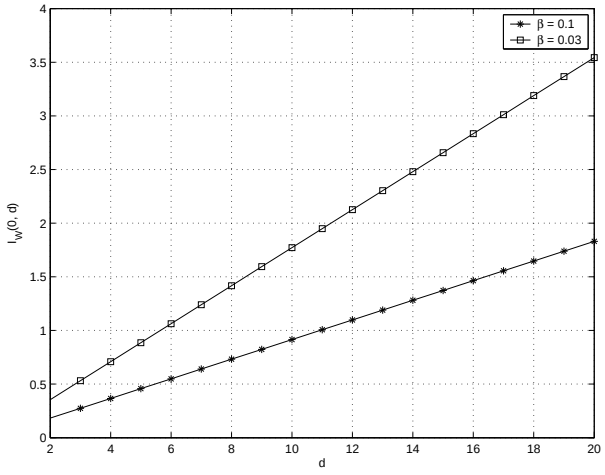

Figure 4.7: Rate function for the multiple link case as a function of d

Substituting in (4.14), for $y = 0$, we have

$$\begin{aligned} & I_{Y_k}(y, d, \bar{B}) = \inf\{I_{\bar{X}}(\bar{x}) : f(\bar{x}) = y\} \\ = & \min_{x_i \in [0,1]} \sum_{i=1}^{d} x_i \log(x_i/p) + (1 - x_i) \log((1 - x_i)/(1 - p)) \\ & \text{such that} \sum_{i=1}^{d} f_i(x_i) = 0\}, \end{aligned} \tag{4.61}$$

and f_i is defined as

$$f_i(x) = \left[\frac{(x - C)_+}{x} M - \min(\bar{B}, (C - x)_+) \right].$$

Note that $f_i(x) = 0$ at C and is strictly increasing and locally concave at C (see Figure 4.5). Also, since the rate function $I_A(x)$ is convex and greater than zero everywhere (except at $x = E[A]$ where $I_A(E[A]) = 0$), if $E[A] \in [C - \bar{B}, C]$ the rate function is a strictly increasing convex function in a small neighbourhood around C. Therefore (4.61) can be written as the convex optimization problem with convex increasing positive cost function $I_A(f^{-1}(z_i))$ under the constraint $\sum_{i=1}^{d} z_i = 0$. From standard optimization theory it follows that, the objective obtains it's minimum when each $z_i = 0$, corresponding to each $x_i = C$. Hence

$$I_{Y_k}(y, d, \bar{B}) = dI_A(C).$$

For our particular example, $I_A(C) = 0.2263$. Hence the probability of packet loss with network coding for this case, scales as $\Theta(\exp(-nd \times (0.2263)))$ showing that coding over larger blocks provides exponential gain in the probability of packet loss. This is analogous to Botvich and Duffield's [50] result for queueing, repeated in (4.1) where $I(b)$ scales linearly as buffer-size b in the large b regime.

We also perform a simulation for the single link case with i.i.d. packet arrivals to each source with a Poisson distribution with mean $E[A_i^m] = 58$, $m = 1, 2, \ldots, n$, $i = 0, 1, \ldots$ and capacity per-flow $C = 60$. We compute the probability of packet loss with queueing in intermediate nodes and *spatial buffer multiplexing* via network coding at the source alone and plot the results in Figure 4.6. We observe that similar performance in terms of packet loss probabilities can be achieved if the number of time-slots over which network coding needs to be performed d is orderwise the same as the buffer b required for queueing.

4.7.2 Path with multiple links

Unlike the single link case, the mapping function $\bar{g}$ for the multiple hop case (see Figure 4.5) is not concave in the neighbourhood of C. However, local properties of the function $\bar{g}(x)$ around $x = C$, allow $I_W(0, d, \bar{B})$ to scale linearly as well with d. However,the analysis is considerably lengthier. Instead, we numerically compute the values of $I_W(0, d, \bar{B})$ for the Bernoulli arrival process in the previous subsection and graphically observe that the rate function does indeed scale linearly with d.

4.8 Numerical results: Single source, large buffer

In the numerical results in Figure 4.8 we compare the effective bandwidth for queueing (see de Veciana and Walrand [57]) against the effective bandwidth for coding (in Eqn (4.56))to achieve the same large buffer QoS asymptotic fall-off δ.

Note that while we only present a sufficient condition for the QoS guarantee, the results in large buffer asymptotics for queueing are both necessary and sufficient. Hence, we can compare our sufficient condition against the tight effective bandwidth condition of de Veciana and Walrand.

We observe that although $C_A(\delta)$ appears to be within a difference of 1 from $\Lambda(\delta)/\delta$, our loss event corresponds to exceeding a delay of d, whereas the de Veciana and Walrand result in Eqn (4.3) [cf. [57]] corresponds to exceeding a queueing buffer of size b. This implies that the coding buffer size required to achieve similar QoS scales as $d\tilde{A}$ for $E[A] \leq$

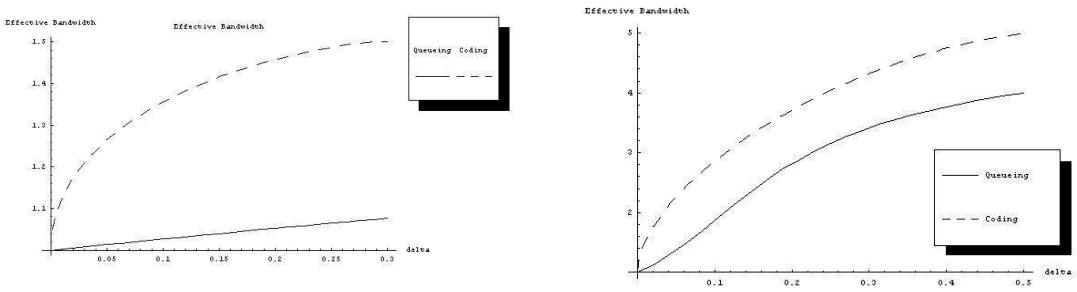

Figure 4.8: Comparing the loss effective bandwidth of queueing vs. coding for for two instances of 2-state Markov Random Sources. $\max_i\{A_i\} = 1.5$ (top), $\max_i\{A_i\} = 5.0$ (bottom), $E[A] = 1$

$\tilde{A} \leq \max_i\{A_i\}$. We conjecture that the 'penalty for coding' $\tilde{A}$ corresponds to the "most likely path" leading to the error event.

These results seem to suggest that buffering is preferable to coding for small networks while our results in 4.5 show that for large networks, the reverse is true.

Chapter 5

TCP-NC in wireless environments

5.1 Introduction

When TCP was designed, it was designed and optimized with the assumption that the networks that it was supposed to operate over have highly reliable node-to-node links so that dropped packets due to bad links are highly unlikely. It was this fact of the wired network that TCP utilized to build a congestion control mechanism; a dropped packet only meant one thing – a buffer overflow due to a congestion somewhere in the network. Thus, when the sender TCP algorithm is notified of lost packets, the additive increase and multiplicative decrease (AIMD) mechanism promptly cut the transmission rate/TCP window size by half. On the other hand, successful reception of window's worth of packets implied under utilization of the network, and thus AIMD mechanism would increase the window size by one packet.

However, a typical wireless link is designed with BER on the order of 10^{-5}, which translates into a packet drop probability of 5-10% for a typical 1KB packet. If TCP is used over the wireless network without any modifications, this considerably reduces the average window size and prevents it from enlarging to any significant portion of the ideal size, the bandwidth-delay product. This results in a small TCP window size and low utilization rate [83, 84]. Further, the asymmetry of the wireless link between uplink and downlink rates causes TCP ACKs congestion on the reverse path, which incorrectly reflects (round trip time) RTT, thereby affecting TCP throughput [85].

To combat such adverse nature of the wireless network, many solutions have been proposed. The commonly deployed solution in UMTS systems involves automatic repeat request (ARQ) between the nodes on the route that TCP connection is made over. ARQ (or Hybrid ARQ) is deployed in a lower layer protocol to deal with packet drops, and hence packet drop due to hostile wireless channels is hidden from the network layer (TCP). However, multiple ARQ requests and the corresponding ACK/NACKs cause retransmission delays that may significantly affect RTT estimation.

Another proposed solution (see [86]) is to code the data stream at a specific coding rate so that the packets can withstand higher drop rate. For example, if the wireless link is known to have 5% drop rate, we can implement coding strategies so that despite 5% loss of packets, we can successfully recover the lost packets from the coding. However, such approach requires the drop rate to be either static or the change in the drop rate be slow enough so that the rate can be fed back to the sender so that he can adjust his coding rate accordingly. Changes in the drop rate can occur over time due to macro scale fading of the wireless channel. For example, if the drop rate changes every RTT, the information about the drop rate will not reach the sender in time to be useful since by the time the information reaches the sender, the drop rate would have changed.

Parallely, we note that there exists a considerable body of literature [87, 88] on modelling of the TCP window process in the presence of active queue management (AQM) systems especially RED [89]. Tinnakornsrisuphap et. al. [88] present a weak limit of the window size process by proving a weak convergence of triangular arrays; the mathematical treatment in this work is based on their treatment. Baccelli et. al. [90] present a fluid limit of the TCP window process, as the number of concurrent flows sharing a link goes to infinity, and the authors show that the deterministic limiting system provides a good approximation for the average queue size and total throughput.

Earlier work on additive increase multiplicative decrease (AIMD) source control used by TCP and its connection to fairness of sharing link bandwidth is considered in [24]; the authors consider a fluid model of the source rate and pose the fair resource allocation problem among heterogeneous users as a convex program. The authors in [91] consider a similar convex program for an end-to-end congestion control scheme in the presence of explicit congestion notification (ECN), which is a proactive congestion avoidance scheme based on packet marking; for their system, the authors prove that there exists a socially fair AIMD scheme to share the link bandwidth. In related work, the use of delay differential models to study internet congestion control using proportionally fair congestion controller (i.e. packet marking based on marking function) for many-flow scenarios is justified in [92].

In this work, we propose a method improving TCP over wireless that does not require feedback (except TCP ACK's) or ARQ. We consider the simple topology of a TCP sender connected via wireline network to an intermediate router and a TCP receiver connected by a wireless channel to this intermediate router. An example scenario would be a cellular access network (such as W-CDMA/WiMax) where the cellular base station is connected to the wired backbone, and only the link between the base station and the mobile

user is wireless.

5.1.1 Main Contributions

In the proposed system, we perform random linear coding (RLC) and priority queueing/transmission at the intermediate router. In our proposed method, the sender encodes the data packets using RLC and sends the data packets along with the coded packets towards the receiver. When the packets (data and coded) arrive at the wireless router, the router transmits only data packets unless there is no data packet to be transmitted in the queue. Thus, data packets have higher priority over coded packets in transmission. When the receiver gets these data and coded packets, the receiver attempts to reconstruct the lost data packets from the successfully received data and coded packets. When the lost data packets can not be reconstructed within time limit, the sender is notified of failed transmissions via duplicate ACK's and timeouts.

Our main contributions are as follows:

(i) For the case where N flows share the router, we formulate a per flow marking process $M^n_{(N)}(t)$ for the n-th flow at RTT-slot t that models both packets lost in the channel due to insufficient coding and packets dropped by the active queue manager (implemented as a marking function) at the router.

(ii) For the AIMD flow control system controlled by the above marking function $M(t)$, we show that as the number of flows sharing the router N tends to infinity, the window size process $W^1_{(N)}(t)$ of each flow converges weakly to a limiting process $W(t)$.

(iii) We upper-bound the limiting marking function $M(t)$; based on this, we prove that the average window size under the limiting distribution is lower bounded by a $1 - 2e^{-1}$ fraction of the ergodic link capacity per flow. This presents an orderwise gain over the performance of native TCP in the presence of random packet errors.

The organization of this chapter is as follows: first, we discuss the system model for a single TCP flow in section 5.2 and describe the additive increase, multiplicative decrease (AIMD) dynamics of a TCP flow. We prove that as the number of flows tends to infinity, the window size process converges weakly (in distribution) to an ergodic process in Section 5.3. Finally, in Section 5.4, we prove that the average window size under the limiting distribution is lower bounded by a $1 - 2e^{-1}$ fraction of the ergodic link capacity per flow.

5.2 System model

This work examines the effect of network-coding on TCP window control, first for a single TCP flow through a simple network consisting of a source, an intermediate buffered wireless router and a receiver, and subsequently, for a generalized model involving multiple TCP flows through the intermediate wireless router. As discussed in Section 5.1, our focus is on understanding hybrid wireline-wireless networks where the TCP source is removed from the wireless hop. This motivates our model where the TCP-source is far removed from the wireless hop, and is therefore unable to respond briskly to variations in the channel conditions.

5.2.1 Single Flow

Consider a TCP connection from source S to destination D going through an intermediate wireline-wireless interface router R. The router receives packets on its wireline interface and transmits packets across a noisy wireless channel to destination D. With slight abuse of notation, we will denote both the buffer and the size of the buffer as B.

Further, for the purpose of analysis, we will assume that each of the TCP connections has the same RTT. We will consider slotted time, with each unit time-slot corresponding to an RTT-interval. In each RTT-interval the wireless router can transmit C packets across the wireless channel, where C is the nominal capacity of the wireless channel. In other words, if the wireless channel experiences no error, then C packets can be transmitted successfully from the wireless router to D. Note that this differs from the information-theoretic capacity for the channel under noiseless conditions – which is infinity.

5.2.1.1 Wireless channel error model

We model error in the channel as a simple i.i.d. packet error process whose parameter remains constant for a block of κ *RTT-intervals* for some $\kappa \in \mathbb{N}$. This is similar to the quasi-static or block-noise model common in wireless communication literature. We index RTT intervals as (i, j) where the index j corresponds to each RTT-interval within a larger block of κ consecutive RTT-intervals, each of which is, in turn, indexed by i.

Let $\{\Omega, \mathcal{F}, \mathbb{P}\}$ be the probability space induced by the packet error parameter process. Let each sample path $\omega \in \Omega$ be written as a sequence $\omega \stackrel{\Delta}{=} \{\omega_{ij}\}_{i=1,j=1}^{i=\infty,j=\kappa}$.

Within each RTT-interval i, j, packets transmitted over the wireless channel suffer degradation due to a Bernoulli error process with parameter $P_i(\omega_{ij}) \in \{p_1, p_2, \ldots, p_\pi\}$ acting

upon each packet over the air independently of other packets in the same RTT-interval.

We remark that the parameter P_i of the Bernoulli packet error process itself is a random variable whose value changes once every κ consecutive RTT-intervals. The probability mass function of the random parameter $P_i(\omega_{ij})$ is specified as $\mathbb{P}(P_i = p_k) \stackrel{\Delta}{=} \tilde{p}_k$, $k = 1, 2, \ldots \pi$; $\sum_{k=1}^{\pi} \tilde{p}_k = 1$.

Although the results in Section 5.3 will hold for the case of any finite κ, we will confine ourselves to the simple case of $\kappa = 1$ to simplify notation. Accordingly, $\omega = \{\omega_t\}_{t=1}^{\infty}$ with the random parameter $P_t(\omega_t) \in \{p_1, p_2, \ldots, p_\pi\}$.

5.2.1.2 Source coding: Random Linear Combination

We will assume that the source receives a stream of A_i user-generated data packets in the i-th RTT-interval and generates a stream of B_i coded packets by using random linear combination (RLC) over the data packets as follows: let each packet $x_{ik}, k = 1, 2, \ldots, A_i$ be represented as an element of some finite field $\mathbb{F}_q$; choose elements $\alpha_{jk} \in \mathbb{F}_q$ uniformly at random and generate B_i coded packets

$$y_{ij} = \sum_{k=1}^{A_i} \alpha_{jk} x_{ik} \tag{5.1}$$

for $j = 1, 2, \ldots, B_i$.

The destination receives the the coefficients of the linear equations, α_{ij}, corresponding to each coded packet as header bits within the packet. Alternately, since in most practical considerations, the coefficients α_{ij} will be generated via a pseudo-random generator, it may be sufficient to initialize the pseudo-random generators at the source and destination to the same state at the beginning of the communication process via some form of handshaking. However, this would require the decoder at the receiver to know the exact number of packets generated in each time-slot so as to maintain both random-number generators at the same state. This information could be encapsulated as part of one or more of the data packets.

If one or more packets are dropped, the receiver recovers the dropped packets by decoding the coded packets received in future time-slots by solving for the unknown values of $x_{i,k}$ from the system of equations in (5.1). Since this is a set of linear equations, the system in (5.1) yields a unique solution provided the random coefficient matrix $\{\alpha_{ij}\}$ is invertible.

Each coded packet, and the corresponding coefficients α_{ij} represent a linear equation over the data packets x_{ik}. The information at the decoder may be represented as a set of

linear equations in known and unknown variables. The known variables correspond to the data packets are directly received by the decoder. The unknown variables are the dropped packets. Hence, the decoder requires as many independent linear equations (coded packets) as the number of unknowns to be able to solve for this set of equations. Note that since the field $\mathbb{F}_q$ is finite, in general, two coded packets have a non-zero probability of being linearly dependent. This corresponds to the event where the matrix of coefficients is singular. In the rest of this work we will loosely refer to the set of linear equations as being *invertible* (uninvertible) if this matrix is not invertible (respectively, not invertible).

However, it is easy to verify that the probability that the coefficient matrix is uninvertible tends to 0 as the size of the finite field F_q tends to infinity. For a value of q that is 20 (30 respectively) bits long, the probability that the coefficient matrix is uninvertible is approximately 10^{-6} ($10^{-8} respectively$) (cf. Chapter 4, Remark 11). Since packets are larger than that, we can neglect the probability that the coefficient matrix is uninvertible. Accordingly, in the rest of the work, we will make the following assumption as a simplification.

Assumption 9. *The set of linear equations in (5.1) is always invertible.*

5.2.1.3 Queue dynamics

A priority rule is implemented at the router R to handle the streams of data and coded packets respectively – we assume that the data packets are first transmitted by the router R. The source maintains a TCP packet window of size $W(t)$ for the t-th RTT-interval containing data packets alone.

The router buffer evolution equation is given by

$$Q(t+1) = \max\{Q(t) + W(t) - C, 0\}, \tag{5.2}$$

leaving the total spare capacity $X(t)$ available for transmitting coded packets at the wireless router, where

$$X(t) \triangleq (C - W)_+ . \tag{5.3}$$

5.2.1.4 TCP window dynamics

The source implements the TCP additive increase multiplicative decrease (AIMD) window algorithm. We will neglect timeouts in our analysis to simplify our model.

The TCP window process at the source is modeled as follows,

$$W(t+1) = \mathbf{1}_{\text{success}}(W(t)+1) + \mathbf{1}_{\text{drop}}\frac{W(t)}{2} \tag{5.4}$$

where the random variable $\mathbf{1}_{\text{success}} \stackrel{\Delta}{=} 1$ when all packets transmitted in the t-th RTT interval have been successfully recovered at the destination; $\mathbf{1}_{\text{drop}} \stackrel{\Delta}{=} 1 - \mathbf{1}_{\text{success}}$ takes the value 1 when either (i) the receiver cannot recover one or more packets corrupted by the packet error process or (ii) there is a tail drop at the router buffer B.

Further, we assume that at any RTT interval a sufficient number of auxiliary coded packets are always available at the router R. Thus, the wireline link $S - R$ is considered to be over-provisioned to accomodate sufficient number of coded packets for the worst channel error parameter. While this is a strong assumption, it is a typical feature of practical network topologies where the end-to-end path capacity of a heterogeneous network is constrained by a bottleneck on the wireless link, whereas the wireline links have large capacities.

We assume full channel state information (CSI) at the router R – in other words, the router knows the value that the random variable P_t takes at each RTT-interval of time.

At the t-th RTT-interval of time, the router R transmits

$$G(P_t, W(t)) = \min\{K, \lceil \alpha W(t)/P_t \rceil\} \tag{5.5}$$

packets over the wireless interface, for some *excess coding factor* $\alpha > 1$. A minimum of K packets will be always transmitted by the wireless encoder for technical reasons which will become clear in Section 5.4.1.

Since the nominal rate of the transmitter is C, we are constrained by $G(P_t, W(t)) \leq C$. The mean number of total (coded and uncoded) packets that need to be transmitted for the destination D to receive $W(t)$ packets can be calculated to be $W(t)/P_t$. We will subsequently show that the excess coding factor creates room for additional coded packets so that the probability that less than $W(t)$ packets are correctly decoded at the destination falls off exponentially fast in link capacity C. This in turn, implies, that for large values of C, small values of $\alpha > 1$ suffice.

Also, the router implements a priority scheduling rule where data packets are transmitted with priority over coded packets.

5.2.1.5 Receiver

The destination receiver D behaves like a regular TCP receiver and transmits an ACK if a packet is correctly received, or transmits a NACK if a packet is corrupted by the error process defined above. The destination D also transmits out of sequence ACKs in the event that a packet is dropped either at the router buffer or in the transmission. In either, case, for modeling purposes, it is the same as a NACK and so we will restrict the destination to only two types of control packets ACK and NACK.

5.2.2 Multiple flows

We consider a sequence of systems indexed by $N \in \mathbb{N}$. For the N-th system, we consider a wireless router R serving multiple flows between source-destination pairs $S_n - D_n$. Like in the preceding subsection, we assume that the path from S_n to R is wireline, whereas the link from R to D_n is wireless. Each link $R - D_n$ is assumed to have independent fading, i.e. each channel has i.i.d. probabilities of error with values from $\mathcal{P}$ as before. We will use the standard assumption that the total link capacity scales with N, i.e. the total channel capacity for the broadcast from R to all the destinations D_n is NC.

5.2.2.1 Queue dynamics

In accordance with the priority rule described in the previous subsection, we assume that the data packets are first transmitted by the router R. Since each source S_n generates $W^n_{(N)}(t)$ packets, the router buffer evolution equation is given by

$$Q_{(N)}(t+1) = \max\{Q_{(N)}(t) + \sum_{n=1}^{N} W^n_{(N)}(t) - NC, 0\}, \tag{5.6}$$

leaving the total spare capacity available at the wireless router given by the expression

$$X_{(N)} = \left(NC - \sum_{n=1}^{N} W^n_{(N)}(t) \right)_+ . \tag{5.7}$$

5.2.2.2 Sharing spare capacity

We will utilize the spare capacity $X_{(N)}$ to transmit the coded packets for the various streams. However, this leads to the important question of how this spare capacity should be divided amongst the various streams, each experiencing varying channel conditions.

The router knows the rates $W^n_{(N)}(t)$/RTT (and therefore can calculate $W^n_{(N)}(t)$ for each stream) as well as the channel error probability P^n_t experienced by each stream n in the t-th RTT interval. We will also assume that the router can estimate the parameter distribution $\{\tilde{p}_j\}_{j=1}^{\pi}$. This implies that the router can estimate the total number of packets

$$\tilde{W}_{(N)}(t) \stackrel{\Delta}{=} \sum_{n=1}^{N} G(P^n_t, W^n_{(N)}(t)), \tag{5.8}$$

to be transmitted over the air, and similarly,

$$\tilde{Q}_{(N)}(t) \stackrel{\Delta}{=} \sum_{n=1}^{N} G(P^n_t, Q^n_{(N)}(t)), \tag{5.9}$$

where $Q^n_{(N)}(t)$ is the number of data packets queued for stream n in the t-th RTT interval such that $Q_{(N)}(t) = \sum_{n=1}^{N} Q^n_{(N)}(t)$. Note that although $Q_{(N)}(t)$ is the size of the buffer occupied by the data packets, the total number (data + coded) of packets, corresponding to the buffered data packets, transmitted over the air is $\tilde{Q}_{(N)}(t)$.

5.3 Multiple flow Analysis

To simplify analysis, we assume that the RTT's for each of the flows passing through the router are the same.

Observe than in the system under consideration, packets are lost when the total number of packets transmitted over the channel $\tilde{W}_N(t)$ exceeds the channel capacity C. Thus, the corresponding active queue management system should seek to alleviate this congestion by marking packets as a function of the spare capacity.

In our model, we will not distinguish between dropping one packet and dropping multiple packets in an RTT; instead assuming that the transmit window will halve at most once in each RTT. We will denote the process that triggers this congestion window back-off by a sequence of $\{0, 1\}$ random variables $M^n_{(N)}(t)$ corresponding to the n-th flow for $n = 1, 2, \ldots, N$ in the t-th RTT-interval, i.e. $M^n_{(N)}(t) = \mathbf{1}_{drop}(n, t)$.

In order to fully specify the model, we need to specify the joint statistics of the random variables $\{M^n_{(N)}(t), W^n_{(N)}(t), P^n_t; i = 1, 2, \ldots, N; t = 1, 2, \ldots\}$. To do so, we first define the i.i.d. random variables $V^n_t; n = 1, 2, \ldots, N; t = 1, 2, \ldots$ where each $V^n_t \sim$ Uniform$[0, 1]$. Note that unlike packet dropping/marking functions implemented in schemes such as RED to monitor queue overflow at the router, where each TCP connection has an individual

marking function associated with it, the packet dropping function in this case takes two parameters – the total occupied link capacity, and the individual flow from each flow. However, as we will show subsequently, the system is designed to keep queue sizes at the buffers very small and hence we do not incorporate queue based dropping at the router buffer.

To implement a simple AQM scheme, we will consider a packet dropping probability function $f^{(N)} : \mathbb{N}_+^2 \to [0, 1]$.

The random variable $A^n_{(N)}(t) \in \{0, 1\}$ which takes the value of 0 when one or more packets from the stream $S_n - D_n$ is dropped by the router in the t-th RTT interval is given by

$$A^n_{(N)}(t) \stackrel{\Delta}{=} \mathbf{1}\left[V^n_t > f^{(N)}(\tilde{W}_{(N)}(t) + \tilde{Q}_{(N)}(t), W^n_{(N)}(t))\right] \tag{5.10}$$

for $n = 1, 2, \ldots, N$; $t = 1, 2, \ldots \infty$, where $\tilde{Q}_{(N)}(t)$ is as defined in (5.9). In other words $A^n_{(N)}(t) = 1$ when the router does not mark/drop packets for stream n at time t.

We will specify the following properties for the random packet dropping function.

Assumption 10. *For some fixed $\Delta \geq 1$*

$$f^{(N)}(Nc, x) = 1, \tag{5.11}$$

for all $c \geq C - \Delta$ and $x \in \mathbb{N}_+$. Also, $f^{(N)}(y, w) = 1$ for all $w > W_{max}$.

Remark 13. *Assumption 10 is the same as assuring that all TCP connections experience window halving when the sum total of the flows (coded + uncoded) exceeds $N(C - \Delta)$.*

Note that this is a conservative marking function since the channel packet error process is random. For small window sizes, it is likely that fewer than $\tilde{W}_{(N)}(t) - W^n_{(N)}(t)$ coded packets are sufficient to compensate for the number of packet losses.

Lemma 22. *Assume that $Q^n_{(N)}(0) = 0$ for all $n = 1, 2, \ldots, N$. Then, for any time t and sample path $\omega \in \Omega$, $Q^n_{(N)} = 0$. This also implies that for all t, $\tilde{Q}_{(N)}(t) = 0$.*

Proof: We will induce over RTT-intervals t. Let us assume that the lemma holds for some t. Then we have, from the queue evolution equation (5.6),

$$Q_{(N)}(t + 1, \omega) = \max\left\{\sum_{n=1}^{N} W^n_{(N)}(t, \omega) - NC, 0\right\}$$

However, Assumption 10 ensures that $\sum_{n=1}^{N} W^n_{(N)}(t) < N(C - \Delta)$, at each RTT-interval t. Hence, $Q_{(N)}(t + 1, \omega) = 0$. Since each $Q^n_{(N)}(t + 1, \omega) \geq 0$, and $Q_{(N)}(t + 1, \omega) = 0$, this

implies that each $Q^n_{(N)}(t+1,\omega) = 0$. From the definition in (5.9) it follows immediately that for all t, $\tilde{Q}_{(N)}(t,\omega) = 0$. ∎

Lemma 22 implies that the TCP-connections are coupled in time, or with each other, only through the window processes $\{W^n_{(N)}(t)\}$ and not through the queue processes $\{Q^n_{(N)}(t)\}$.

The next assumption specifies a technical constraint on the sequence of marking functions $f^{(N)}(\cdot)$ to ensure convergence in Theorem 8.

Assumption 11. *There exists a continuous function $f : \mathbb{R}^2_+ \to [0,1]$ such that for each $N \in \mathbb{N}$,*

$$f^{(N)}(x,y) = f(N^{-1}x, y).$$

Further, for all the packets that are transmitted, on flow $S_n - D_n$ in RTT-interval t, let C^n_t be the required number of coded packets transmitted by the router. Thus,

$$C^n_t = G(P^n_t, W^n_{(N)}(t)) - W^n_{(N)}(t) \tag{5.12}$$

when event $\mathcal{J} \stackrel{\Delta}{=} \{f^{(N)}(\tilde{W}_{(N)}(t) + \tilde{Q}_{(N)}(t), W^n_{(N)}(t)) < 1\}$ occurs, i.e. when the channel does not experience overflow. Also, let us define

$$H^n_j(t) = \begin{cases} 1 & \text{if the j}-\text{th packet is received correctly} \\ 0 & \text{if the j}-\text{th packet is corrupted by error.} \end{cases}$$

Let us further define the r.v. $\mathcal{B}^n_{(N)}(t)$ corresponding to the event that the number of received coded packets are not sufficient to decode the number of data packets corrupted by the channel. Also, let

$$\hat{P}^n_t \stackrel{\Delta}{=} \frac{\sum_{j=1}^{G(P^n_t, W^n_{(N)}(t))} H^n_j(t)}{G(P^n_t, W^n_{(N)}(t))} \tag{5.13}$$

be the empirical fraction of packets (coded + uncoded) correctly received by the destination D_n in the t-th RTT interval when the channel error parameter is P^n_t. Let us define

$$B^n_{(N)}(t) = \mathbf{1}\left\{G(P^n_t, W^n_{(N)}(t))\hat{P}^n_t > W^n_{(N)}(t)\right\}. \tag{5.14}$$

The Bernoulli random variable $\bar{B}^n_{(N)}(t) \in \{0,1\}$ which takes the value 1 when all data packets transmitted by the router in $W^n_{(N)}(t)$ are correctly decoded by the receiver,

can be represented as

$$\begin{aligned}
& \bar{B}^n_{(N)}(t) \\
= \; & \mathbf{1}\left\{\sum_{j=1}^{C^n_t} H^n_j(t) > \sum_{k=C^n_t+1}^{G(P^n_t, W^n_{(N)}(t))} (1 - H^n_k(t))\right\} \\
= \; & \mathbf{1}\left\{\sum_{j=1}^{C^n_t} H^n_j(t) > W^n_{(N)}(t) - \sum_{k=C^n_t+1}^{G(P^n_t, W^n_{(N)}(t))} H^n_k(t)\right\} \\
= \; & \mathbf{1}\left\{\sum_{j=1}^{G(P^n_t, W^n_{(N)}(t))} H^n_j(t) > W^n_{(N)}(t)\right\}
\end{aligned}$$

Observe that in our model, packets are successfully received by the receiver (and thus there are no drops and, in turn, window size increases) if both $\{A^n_{(N)}(t) = 1\}$ and $\{\bar{B}^n_{(N)}(t) = 1\}$. We can then represent the *marking event* $\{M^n_{(N)}(t) = 1\}$, which corresponds to the event that the data packet window is halved, as follows

$$\begin{aligned}
& \{M^n_{(N)}(t) = 1\}^c \\
= \; & \{A^n_{(N)}(t) = 1\} \cap \{\bar{B}^n_{(N)}(t) = 1\} \\
\overset{(a)}{=} \; & \{A^n_{(N)}(t) = 1\} \cap \{\bar{B}^n_{(N)}(t) = 1\} \cap \mathcal{J} \\
\overset{(b)}{=} \; & \{A^n_{(N)}(t) = 1\} \cap \mathbf{1}\left\{\sum_{j=1}^{G(P^n_t, W^n_{(N)}(t))} H^n_j(t) > W^n_{(N)}(t)\right\} \\
= \; & \{A^n_{(N)}(t) = 1\} \cap \mathbf{1}\left\{G(P^n_t, W^n_{(N)}(t))\hat{P}^n_t > W^n_{(N)}(t)\right\} \\
\overset{(c)}{=} \; & \{A^n_{(N)}(t) = 1\} \cap \{B^n_{(N)}(t) = 1\}
\end{aligned}$$

where (a) follows from the observation that $\{A^n_{(N)}(t) = 1\} \subseteq \mathcal{J}$ according to the definition of marking function in Assumption 10; (b) follows from the relation in (5.12) and the expression for $\bar{B}^n_{(N)}(t)$ above; and (c) follows from (5.14).

Thus, we can represent the marking event random variable

$$M^n_{(N)}(t) = 1 - A^n_{(N)}(t)B^n_{(N)}(t). \tag{5.15}$$

Let $\mathcal{F}_t$ be the filtration adapted to the process $\{W^n_{(N)}(t), P^n_t, n = 1, 2, \ldots, \}$ for $t = 0, 1, 2, \ldots, \infty$.

Remark 14. *Note that the random variables $A^n_{(N)}(t)$ and $B^n_{(N)}(t)$ are not independent from each other since they share the common random variable $W^n_{(N)}(t)$. Thus, despite the fact that the error process in the channel that corrupts the packets is independent of the random variable V^n_t associated with the random variable $A^n_{(N)}(t)$, the actual* probability *that data packets on the n-th stream are decoded correctly in the t-th RTT-interval depends on the window size $W^n_{(N)}(t)$ and thus the variables are not independent. However, conditioned on $\mathcal{F}_t$,*

$$\begin{aligned} & \mathbb{P}(A^n_{(N)}(t)B^n_{(N)}(t)|\mathcal{F}_t) \\ = \quad & \mathbb{P}(A^n_{(N)}(t)|\mathcal{F}_t)\mathbb{P}(B^n_{(N)}(t)|\mathcal{F}_t), \end{aligned} \tag{5.16}$$

i.e. $A^n_{(N)}(t)$ and $B^n_{(N)}(t)$ are independent, conditioned on $\mathcal{F}_t$.

Accordingly, we can now represent the evolution of the window by the following stochastic dynamic equation:

$$W^n_{(N)}(t+1) = W^n_{(N)}(t) + 1 - M^n_N(t)\left\lfloor 1 + \frac{W^n_{(N)}(t)}{2} \right\rfloor. \tag{5.17}$$

Equation (5.17), together with the queue evolution equation in (5.6), window resizing variable (5.15), and the definition of $W_{(N)}(t)$ from (5.8) completely specify the dynamics of the system under consideration.

We will next make a technical assumptions to ensure convergence in Theorem 8. We specify that the initial window size when the system starts is chosen uniformly at random, and also that initially, the router queue is empty.

Assumption 12. *For each $N \in \mathbb{N}$, the initial state of the packet windows and router queue are given by*

$$Q_{(N)}(0) = 0, \text{ and } W^n_{(N)}(0) = \gamma C$$

for some $\gamma \sim$ Uniform$(0,1)$.

Consider any RTT-interval indexed by a finite positive integer t. In the remainder of this section, we will prove a weak convergence for the window size process $W^n_{(N)}(t)$ as $N \to \infty$. The structure of the lemmata is similar to the construction in [88] where the authors prove a weak Law of Large Numbers for the triangular array, $\{W^n_{(N)}(t) \ \ , n = 1, 2, \ldots, N\}$ of TCP windows with packet drops in wireline networks.

However, our model and proofs differ from [88] in two respects. First, while the link capacity is kept constant for all RTT intervals in [88], we consider the case where the channel packet drop probability (and therefore the information theoretic capacity) of the link varies from RTT-interval to RTT-interval. Our proofs consider this aspect, and are therefore significantly different from their counterparts in [88]. Secondly, the marking function we consider is different from [88], thus requiring a different proof.

Theorem 8. *Under the conditions of Assumptions 11 and 12, then for each $t = 0, 1, \ldots$, the following limits hold* [1]

$$[A{:}t] \qquad \frac{\tilde{W}_{(N)}(t)}{N} \rightarrow_N^P \tilde{w}_{tot}(t) \tag{5.18}$$

$$[B{:}t] \quad W_{(N)}^1(t) \Rightarrow_N W(t) \tag{5.19}$$

For $k = 1, 2, \ldots, \pi$, let $I_k(t) \subseteq \{1, 2, 3, \ldots, N\}$ be the set of source-destination indexes such that $P_t^n = p_k$, $\forall k \in I_k(t)$. Let J_n^k be a Bernoulli random variable taking $J_n^k = 1$ when $P_t^n = p_k$ (i.e. when $n \in I_k(t)$) and 0 otherwise. For any function $g : \mathbb{R}_+ \rightarrow R$ and any $k = 1, 2, \ldots, \pi$,

$$[C{:}t] \quad \frac{1}{N} \sum_{n=1}^{N} J_n^k g(W_{(N)}^n(t)) \rightarrow_N^P \tilde{p}_k E[g(W(t))]. \tag{5.20}$$

Further, $W_{(N)}^n(t)$, $n = 1, 2, \ldots, N$ is asymptotically independent which is defined as follows: for any finite subset of flows indexed by $\mathfrak{I} \subset \mathbb{N}$ and any set of values $\{a_m \in \mathbb{N} | m \in \mathfrak{I}, 0 < a_m < W_{\max}\}$

$$[D{:}t] \quad \lim_{N \rightarrow \infty} \mathbb{P}\Big(\bigcap_{m \in \mathfrak{I} \cap [1,N]} \{W_{(N)}^m(t) = a_m\} \Big) = \prod_{m \in \mathfrak{I}} \mathbb{P}(W(t) = a_m). \tag{5.21}$$

Moreover, the resulting limiting processes are given by

$$W(t+1) \quad = \quad W(t) + 1 - M(t) \left\lfloor 1 + \frac{W(t)}{2} \right\rfloor \tag{5.22}$$

where $M(t) : \mathbb{N} \times \Omega \rightarrow [0, 1]$ is an appropriately defined marking function, and

$$\tilde{w}_{tot}(t) = \alpha E[W(t)](E[1/P_t^1])^{-1}. \tag{5.23}$$

Proof: Without loss of generality, from Assumptions 12 and 11 we can immediately see that [A:0], [B:0], [C:0] and [D:0] hold. We will now induce over the set of natural numbers. Let our induction hypothesis be that [A:t], [B:t], [C:t] and [D:t] hold. We will now show that [A:t+1], [B:t+1], [C:t+1] and [D: t+1] hold as well.

[1] Notation: $\rightarrow_N^P$ denotes in probability as $N \rightarrow \infty$; $\Rightarrow_N$ denotes convergence in distribution as $N \rightarrow \infty$.

Lemma 23. *[A:t] and [B:t] $\longrightarrow$ [B: t+1]*

Proof: See Appendix 5.5.1. ∎

Lemma 24. *[A:t], [B:t], [D:t] $\longrightarrow$ [D:t+1]*

Proof: This result is analogous to the corresponding asymptotic independence result in [88]; we present it in Appendix 5.5.2 only for sake of completeness. ∎

Lemma 25. *[C:t+1] $\longrightarrow$ [A:t+1]*

Proof: From (5.8),

$$\begin{aligned}\frac{\tilde{W}_{(N)}(t+1)}{N} &= \frac{1}{N}\sum_{n=1}^{N}\frac{W^n_{(N)}(t+1)\alpha}{P^n_t} \\ &= \sum_{k=1}^{\pi}\frac{\alpha}{p_k}\left(\frac{1}{N}\sum_{n\in I_k}W^n_{(N)}(t)\right). \end{aligned} \tag{5.24}$$

Observe that from the definition of J^k_n in [C:t+1] (5.20),

$$\frac{1}{N}\sum_{n\in I_k}W^n_{(N)}(t) \rightarrow^P_N \tilde{p}_k E[W(t+1)] \tag{5.25}$$

for each $k = 1, 2, \ldots, \pi$.

Recall from the system model in Section 5.2 that the channel packet drop rate P^1_t for the first flow is distributed as $\mathbb{P}(P^1_t = p_k) = \tilde{p}_k$. Since the number of channel error parameters, π, is a finite constant, it follows from (5.24) and (5.25) that

$$\begin{aligned}\frac{\tilde{W}_{(N)}(t+1)}{N} &\rightarrow^P_N \alpha E[W(t+1)]\sum_{k=1}^{\pi}\frac{1}{p_k}\tilde{p}_k \\ &= \alpha E[W(t)]E[1/P^1_t]\end{aligned}$$

where the last relation follows since for any flow, the channel error process is i.i.d. and distributed $\sim P^1_t$.

∎

Lemma 26. *[B:t+1], [D:t+1] $\longrightarrow$ [C:t+1]*

Proof: Observe that by our assumption each $W^n_{(N)}(0)$ is chosen uniformly at random from the set of possible window sizes, and thus the window processes $W^n_{(N)}(t+1)$ are exchangeable. This, implies that for any mapping $g : \mathbb{N} \to \mathbb{R}$,

$$\begin{aligned}
& \text{var}\left[\frac{1}{N}\sum_{n=1}^{N} J^k_n g(W^n_{(N)}(t+1))\right] \\
= \; & N^{-2}\sum_{n=1}^{N}\text{var}[g(W^n_{(N)}(t))J^k_n] + N^{-2}\times \\
& \sum_{m,n=1,m\neq n}^{N}\text{cov}[g(W^n_{(N)}(t+1))J^k_n, g(W^m_{(N)}(t+1))J^k_m] \\
= \; & N^{-1}\text{var}[g(W^n_{(N)}(t+1))J^k_1] \qquad (5.26)\\
& +\frac{N-1}{N}\text{cov}[g(W^1_{(N)}(t+1))J^k_1, g(W^2_{(N)}(t+1))J^k_2]
\end{aligned}$$

Now, since the window evolution process $W^1_{(N)}(t)$ is independent of the channel error paramter process (the process by which the channel noise probability p_k changes from RTT-interval to RTT-interval), the random variables $W^1_{(N)}(t+1)$ and J^k_1 are independent. Also, since $W^1_{(N)}(t+1) < W_{\max}$, $\text{var}[g(W^n_{(N)}(t+1))]$ is finite and $J^k_1 \sim$ Bernoulli$\{0,1\}$), $\text{var}[g(W^1_{(N)}(t+1))J^k_1]$ is finite. Hence, $N^{-1}\text{var}[g(W^1_{(N)}(t+1))J^k_1] \to_N 0$ for any sample path.

We will next show that since $W^1_{(N)}(t+1)$ and $W^2_{(N)}(t+1)$ are asymptotically independent (from statement [D:t]), and the random variables J^k_1, J^k_2 are independent of each other and independent to the window processes $W^1_{(N)}(t+1), W^2_{(N)}(t+1)$, the covariance term in (5.26) tends to 0 as $N \to \infty$.

Consider any arbitrary functions $g_1, g_2 : \mathbb{N} \to \mathbb{R}$. Then, from [D:t] we have,

$$\begin{aligned}
& \lim_{N\to\infty} E[g_1(W^1_{(N)}(t+1))g_2(W^2_{(N)}(t+1))] \\
& = E[g_1(W(t+1))]E[g_2(W(t+1))]. \qquad (5.27)
\end{aligned}$$

Now, since $J^k_1, J^k_2 \in \{0,1\}$ using the asymptotic independence property of (5.27),

$$\begin{aligned}
& \lim_{N\to\infty} E\left[g(W^1_{(N)}(t+1))J^k_1 g(W^2_{(N)}(t+1))J^k_2\right] \\
= \; & E\left[g(W^1_{(N)}(t+1))g(W^2_{(N)}(t+1))\right]\mathbb{P}(J^k_1=1)\mathbb{P}(J^k_2=1) \\
= \; & E[g_1(W(t+1))]E[g_2(W(t+1))]\mathbb{P}(J^k_1=1)\mathbb{P}(J^k_2=1) \\
= \; & E[g(W(t+1))J^k_1]E[g(W(t+1))J^k_2]
\end{aligned}$$

where the last relation follows from the fact that $J_1^k \perp J_2^k \perp W(t+1)$.

The above equation implies that as $N \to \infty$, $\mathrm{cov}[g(W^1_{(N)}(t+1))J_1^k, g(W^2_{(N)}(t))J_2^k] \to 0$ and thus,

$$\lim_{N\to\infty} \mathrm{var}\left[\frac{1}{N}\sum_{n=1}^{N} J_n^k g(W^n_{(N)}(t+1))\right] = 0.$$

Hence, using Chebyshev's inequality, we conclude that

$$\begin{aligned}\lim_{N\to\infty} \frac{1}{N}\sum_{n=1}^{N} J_n^k g(W^n_{(N)}(t+1)) &= E[J_1^k g(W^1_{(N)}(t+1))] \\ &= \tilde{p}_k E[g(W(t+1))]\end{aligned}$$

where the last expression follows from [B:t] and $J_1^k \perp W^1_{(N)}(t+1)$. ∎

Proof [Theorem 8]: Assume the result in Theorem 8 holds at RTT-interval t. From Lemma 23, [A:t] and [B:t] imply that [B:t+1] holds. Next, since [A:t],[B:t] and [D:t] hold, from Lemma 24, we see that [D:t+1] is satisfied.

Since [B:t+1] and [D:t+1] hold, Lemma 26 implies that [C:t+1] will hold as well. Finally, from Lemma 25, [C:t+1] implies [A:t+1] thus proving that all the statements on Theorem 8 hold at $t+1$ if they hold at t. Since the statements trivially hold for time $t = 0$, by the principle of mathematical induction, we are now done. ∎

5.4 Fixed point for the window evolution equations

To be able to evaluate the "efficiency" of our transmission scheme, we need to understand the steady state performance of the limiting window evolution (5.22). However, note that we first need to establish that the window evolution process $\{W(t)\}$ is ergodic.

Looking at the equations (5.22),(5.23), we immediately see that the process $\{W(t)\}$ is a discrete-time Markov chain where the state transitions are driven by the uniform random variables $(V(t), V'(t))$.

Corollary 2. *The window evolution process $\{W(t)\}$ is ergodic with some limiting distribution W^*.*

Proof: Further, since $W(t) \le W_{max}$ for all integers $t \ge 0$, the Markov chain can also be seen to be finite. Also, we note from Assumption 10 that for all $t \ge 0$, there exists a finite probability $\epsilon_p > 0$, strictly greater than zero, that the window size $W(t)$ halves in the next

time-slot – this implies that the Markov chain is irreducible. Finally, the random variables $(V(t), V'(t))$ ensure that the Markov chain is aperiodic. It follows immediately that $\{W(t)\}$ is ergodic. ∎

Remark 15. *Observe that although the process $\{W(t)\}$ as defined by equation (5.22)is proved to be ergodic in the corollary above, we must point out as a caveat that it does not immediately imply that the limiting proces of $W^n_{(N)}(t)$ for some finite $n < N$ converges to an ergodic process. This is because the limit results in Theorem 8 apply only for* any finite *value of t, while taking N to infinity.*

However, we will approximate the limiting distribution of $W^n_{(N)}(t)$ as $W(t)$ even as $t \to \infty$. This can be thought of an exchange of limits from time and flow; a similar treatment is presented elsewhere in the literature in [61, 93].

5.4.1 Stationary Distribution of $W(t)$

In the following, we will provide a closed form expression for the lower bound on the ergodic average of the process $W(t)$ governed by equation (5.22). Since $W(t)$ is ergodic, the time-average of the window size $W(t)$ will tend to the stationary average distribution of the Markov chain governed by equation equation (5.22) as $t \to \infty$. To determine this stationary average, we will assume that the finite field Markov chain governed by equation (5.22) starts at the stationary distribution at time $t = 0$. Note that we do not require the real system as described in Section 5.3 to start at the stationary distribution at $t = 0$; it is only for the purpose of analysis that we make the initial stationary distribution assumption. This implies that for all $t > 0$, $E[W(t)] = E[W(t+1)] = E[W]$.

In the remainder of this section, we will confine our discussion, without loss of generality, to the limiting flow corresponding to the TCP connection with index 1.

Observe that the probability of the event $\{B^n_{(N)}(t) = 1\}$ corresponds to the binomial distribution with parameters (N, P^1_t) and is easier expressed in terms of a tight Chernoff bound as $N \to \infty$. Since excess coding factor $\alpha > 1$, any $W^n_{(N)}(t) = w$, $w \in 1, 2, \ldots, W_{\max}$,

$P_t^1 = p_k$

$$\begin{aligned}
& \mathbb{P}(B_{(N)}^1(t) = 0 | W_{(N)}^n(t) = w, P_t^1 = p_k) \\
= \; & \mathbb{P}\left(\left\lceil \frac{w\alpha}{p_k} \right\rceil \frac{\sum_{j=1}^{G(p_k,w)} H_j^1(t)}{G(p_k,w)} < w \right) \\
\leq \; & \mathbb{P}\left(\frac{w\alpha}{p_k} \frac{\sum_{j=1}^{G(p_k,w)} H_j^1(t)}{G(p_k,w)} < w \right) \\
= \; & \mathbb{P}\left(\frac{\sum_{j=1}^{G(p_k,w)} H_j^1(t)}{G(p_k,w)} < \frac{E[\frac{\sum_{j=1}^{G(p_k,w)} H_j^1(t)}{G(p_k,w)}]}{\alpha} \right) \\
\leq \; & \exp(-D(p_k/\alpha || p_k) G(p_k,w))
\end{aligned}$$

where $D(x||y) \stackrel{\Delta}{=} x\log(\frac{x}{y}) + (1-x)\log(\frac{1-x}{1-y})$ is the Kullback-Leibler distance between Bernoulli distributions with parameters x and y respectively, and the empirical probability of packet loss $\hat{P}_t^1$ has mean $E[\hat{P}_t^1] = p_k$. Since the set $p_1, p_2, \ldots, p_\pi$ is finite, we can write the bound

$$\begin{aligned}
& \mathbb{P}(B_{(N)}^1(t) = 0 | W_{(N)}^n(t) = w, P_t^1 = p_k) \\
\leq \; & e^{-\min_{p_k} D(p_k/\alpha || p_k) G(p_k,w)}.
\end{aligned} \tag{5.28}$$

In order to lower-bound the number of packets that a flow transmits, we will consider the following design modification: the minimum number of coded+uncoded packets transmitted, per flow, by the wireless transmitter is $K < C$. Hence, the effective total number of packets transmitted by the wireless transmitter is $\max\{K, G(p_k, w)\}$ for stream 1 at time-slot t. Thus, even in that case where the window size W is small, the number of packets (coded+uncoded) transmitted for each flow is at least K.

Accordingly, we may write the above equation as

$$\begin{aligned}
& \mathbb{P}(B_{(N)}^1(t) = 0 | W_{(N)}^n(t) = w, P_t^1 = p_k) \\
& \leq \min\{e^{-\min_{p_k} D(p_k/\alpha || p_k) G(p_k,w)}\} \\
& \leq \epsilon_K
\end{aligned} \tag{5.29}$$

for some suitable value of $\epsilon_K > 0$, where the last bound is as a result of the finite set of values that w takes. Also, note that the bound in (5.29) is uniform over all values that $W_{(N)}^n(t)$ may take. Further, as C and K take larger values, and the value of the excess coding factor α is made larger, ϵ_K can be made smaller for any value of w and p_k.

However from (5.54) we note that,

$$\begin{aligned} & Y^1_{(N)}(t) \\ = \quad & 1 - \mathbb{P}(B^1_{(N)}(t) = 0 | \mathcal{F}_t) \\ = \quad & 1 - \mathbb{P}(B^1_{(N)}(t) = 0 | W^n_{(N)}(t), P^1_t) \qquad (5.30) \\ \Rightarrow_N \quad & Y^1(t). \end{aligned}$$

where (5.30) follows from the observation that $W^n_{(N)}(t), P^1_t$ are sufficient statistics for $B^1_{(N)}(t)$ according to the definition of $B^1_{(N)}(t)$ in (5.14).

Hence (5.30), together with the bound in (5.29), implies that

$$Y^1(t) \geq 1 - \epsilon_K. \qquad (5.31)$$

In the following lemma, we will bound the value of ϵ_K.

Lemma 27. *Equation (5.31) is satisfied by* $\epsilon_K = e^{-K \min_{p_k} D(p_k/\alpha || p_k)}$.

Proof: We repeat (5.29) below,

$$\begin{aligned} & \mathbb{P}(B^1_{(N)}(t) = 0 | W^n_{(N)}(t) = w, P^1_t = p_k) \\ & \leq \min\{e^{-\min_{p_k} D(p_k/\alpha || p_k) G(p_k, w)}\} \end{aligned}$$

Noting from (5.5) that the minimum value of $G(p_k, w)$ is K, the result follows immediately. ∎

Now, for the 1-st flow, where $W^1(t)$ is the limiting window process distributed as $W(t)$ in Theorem 8, we can write

$$\begin{aligned} & \mathbb{P}(M^1(t) = 1 | W^1(t) = w) \\ = \quad & E[M^1(t) | W^1(t) = w] \\ = \quad & E[E[M^1(t) | \mathcal{F}_t] | W^1(t) = w] \\ = \quad & E[1 - E[A^1(t) | \mathcal{F}_t] E[B^1(t) | \mathcal{F}_t] | W^1(t) = w] \\ = \quad & E[1 - Z^1(t) Y^1(t) | W^1(t) = w] \qquad (5.32) \\ = \quad & [1 - (1 - f(\tilde{w}_{tot}(t), w)) E[Y^1(t) | W^1(t) = w]] \qquad (5.33) \end{aligned}$$

where (5.32) follows from the limits in (5.55) and (5.56).

Recall from (5.23) that $\tilde{w}_{tot} = \frac{1}{\alpha} E[W^1(t)](E[1/P^1_t])^{-1}$ and from the stationary ergodicity property in Corollary 2 that $E[W^1(t)] = E[W]$, i.e. the window size distribution

is invariant over RTT-slots t. Therefore, since $(W^1_{(N)}(t), Z^1_{(N)}(t)) \Rightarrow_N (W^1(t), Z^1(t))$ from (5.56), $Z^1(t)$ is measurable with respect to $W^1(t)$. This explains the separation in (5.33).

As an aside, we must point out that by the definition in (5.55) $Y^1_{(N)}(t)$, and therefore $Y^1(t)$, is measurable with respect to the tuple $(W^1(t), P^1_t)$. Thus the expectation in $E[Y^1(t)|W^1(t) = w]$ is over the channel error distribution P^1_t.

Noting that $W^1(t) \sim W(t)$ (from Corollary 2), we can define an effective marking function $f_{eff} : \mathbb{R} \times \mathbb{N} \to [0, 1]$ as follows:

$$\begin{aligned} & f_{eff}(E[W], W^1(t)) \\ \triangleq \;& 1 - (1 - f(\tilde{w}_{tot}(t), W^1(t)))E[Y^1(t)|W^1(t)], \end{aligned} \tag{5.34}$$

so that $\mathbb{P}(M^1(t) = 1|W^1(t)) = f_{eff}(E[W], W^1(t))$.

In addition to the conditions on the marking function f made in Section 5.3, we will also make the following monotonicity assumption on f.

Assumption 13. *The function $f(x, y)$ increases in variable x.*

Remark 16. *Note that in equation (5.10) the marking function $f^{(N)}()$ takes $\tilde{W}_{(N)}(t)$ – which is the total flow (coded and data packets)– as its first parameter. As such, by Assumption 11, the parameter x reflects this total load on the outgoing shared channel scaled by $1/N$, where N is the total number of flows. This implies that Assumption 13 reflects the marking function design principle that packets are more likely to be dropped if the overall load on the shared channel is higher.*

In the following analysis, we will confine ourselves to flow 1. Also, for ease of notation, we will drop the superscript from $W^1(t), M^1(t), Y^1(t), Z^1(t)$ and P^1_t; to $W(t), M(t), Y(t), Z(t)$ and P_t respectively, in the remainder of this section.

Let us define the function

$$\begin{aligned} & \phi(E[W], W(t)) \\ \triangleq \;& \min\{(1 - f(\tilde{w}_{tot}(t), W(t)))(\epsilon_K - (1 - E[Y(t)|W(t)])) \\ & + f(\tilde{w}_{tot}(t), W(t))\epsilon_K, (1 - f(\tilde{w}_{tot}(t), W(t))E[Y(t)|W(t)]\} \\ \geq \;& 0 \end{aligned} \tag{5.35}$$

where (5.35) follows from (i) (5.31) and (ii) the property that $f : \mathbb{R}^2 \to [0, 1]$ in Assumption 11. Note that the function $\phi(E[W], W(t))$ is non-negative for all pairs $E[W], W(t)$.

We can then define a "stronger" marking random variable $\hat{M}(t)$, and a corresponding effective marking function $\hat{f}_{eff}$ as follows:

$$\begin{aligned} & \mathbb{P}(\hat{M}(t)=1|W(t)) \\ = \;& \hat{f}_{eff}(E[W],W(t)) \\ \stackrel{\Delta}{=} \;& \min\{f(\tilde{w}_{tot}(t),W(t))+\epsilon_K,1\} \qquad (5.36) \\ = \;& f_{eff}(E[W],W(t))+\phi(E[W],W(t)) \qquad (5.37) \end{aligned}$$

where (5.37) follows from the definition of $\phi()$ upon rearranging terms.

Also, noting that $\hat{M}(t)$ is measurable with respect to $W(t)$, the effective marking function satisfies

$$\hat{M}(t)=\mathbf{1}\{U(t)>\hat{f}_{eff}(E[W],W(t))\} \qquad (5.38)$$

where $U(t)\sim$ Uniform$[0,1]$ i.i.d. for each RTT slot t.

Let $V(t)$ be the random process that satisfies the recursion

$$V(t+1)=V(t)+1-\hat{M}(t)\left\lfloor 1+\frac{V(t)}{2}\right\rfloor. \qquad (5.39)$$

$\{V(t)\}$, as governed by the recursion above, is a discrete time discrete space Markov Chain over a finite irreducible state-space. Hence, using arguments similar to Corollary 2, we can claim that $\{V(t)\}$ is ergodic, with a limiting distribution V^*.

Lemma 28. $E[V^*]\leq E[W^*]$

Proof: Since W^* is the limiting distribution to (5.22), we can define the corresponding limiting random marking function M^* satisfying

$$\mathbb{P}(M^*=1|W^*)=f_{eff}(E[W^*],W^*).$$

Let $g:\mathbb{N}\to\mathbb{R}$ be any bounded measurable function. Then, since W^* satisfies

$$W^*=W^*+1-M^*\left\lfloor 1+\frac{W^*}{2}\right\rfloor$$

$$\begin{aligned} E[g(W^*)] &= E[g(W^*+1-M^*\left\lfloor 1+\frac{W^*}{2}\right\rfloor] \\ &= E[g(W^*+1)|M^*=0]\mathbb{P}(M^*=0) \\ &\quad +E[g(\lceil\frac{W^*}{2}\rceil)|M^*=1]\mathbb{P}(M^*=1) \qquad (5.40) \end{aligned}$$

But,

$$\begin{aligned} & E[g(W^*)|M^*=1] \\ = & \sum_{w=1}^{W_{\max}} g(w)\mathbb{P}(W^*=w|M^*=1) \\ = & \frac{1}{\mathbb{P}(M^*=1)}\sum_{w=1}^{W_{\max}} g(w)\mathbb{P}(W^*=w)\mathbb{P}(M^*=1|W^*=w) \\ = & \frac{1}{\mathbb{P}(M^*=1)}\sum_{w=1}^{W_{\max}} g(w)\mathbb{P}(W^*=w)f_{eff}(E[W^*],W^*) \\ = & \frac{1}{\mathbb{P}(M^*=1)}E_{W^*}[g(W^*)f_{eff}(E[W^*],W^*)]. \end{aligned}$$

This implies that

$$\begin{aligned} & E[g(\lceil\frac{W^*}{2}\rceil)|M^*=1]\mathbb{P}(M^*=1) \\ = & E_{W^*}[g(\lceil\frac{W^*}{2}\rceil)|f_{eff}(E[W^*],W^*)]. \end{aligned} \tag{5.41}$$

Similarly, we can write

$$\begin{aligned} & E[g(W^*+1)|M^*=0]\mathbb{P}(M^*=0) \\ = & \sum_{w=1}^{W_{\max}} g(w+1)\mathbb{P}(W^*=w)\mathbb{P}(M^*=0|W^*=w) \\ = & \sum_{w=1}^{W_{\max}} g(w+1)\mathbb{P}(W^*=w)(1-f_{eff}(E[W^*],W^*)) \\ = & E_{W^*}[g(W^*+1)(1-f_{eff}(E[W^*],W^*))]. \end{aligned}$$

Substituting the above relation and (5.41) into equation (5.40) and collecting terms, we can then write

$$\begin{aligned} E_{W^*}[g(W^*+1)-g(W^*)] = E_{W^*}[\{g(W^*+1) - \\ g(\lceil\frac{W^*}{2}\rceil)\}f_{eff}(E[W^*],W^*)]. \end{aligned} \tag{5.42}$$

Similarly, for any mapping $g' : \mathbb{N} \to \mathbb{R}$, we have

$$\mathbb{P}(\hat{M}^*=1|V^*) = \hat{f}_{eff}(E[V^*],V^*),$$

and thus,

$$E_{V^*}[g'(V^*+1) - g'(V^*)] = E_{V^*}[\{g'(V^*+1) - g'(\lceil \frac{V^*}{2} \rceil)\}\{f_{eff}(E[V^*], V^*) + \phi(E[V^*], V^*)\}], \tag{5.43}$$

where $\phi()$ is as defined in (5.35).

Further, since the Markov Chain defined by the recursion in (5.39) is finite state, aperiodic and irreducible $\mathbb{P}_{V^*}(x) > 0$ for $x \in \{1, 2, \ldots, W_{\max}\}$, where $\mathbb{P}_{V^*}()$ is the probability measure corresponding to the distribution V^*. We can therefore define the function $g' : \mathbb{N} \rightarrow \mathbb{R}$ as follows

$$g'(x) \stackrel{\Delta}{=} g(x)\frac{\mathbb{P}_{W^*}(x)}{\mathbb{P}_{V^*}(x)}.$$

By the above 'change in measure',

$$E_{V^*}[g'(A)] = E_{W^*}[g(A)]$$

for any set $A \subseteq \{1, 2, \ldots, W_{\max}\}$.

Thus, we can write (5.43) as

$$\begin{aligned} & E_{W^*}[g(W^*+1) - g(W^*)] \\ = \; & E_{W^*}[\{g(W^*+1) - g(\lceil \frac{W^*}{2} \rceil)\} f_{eff}(E[V^*], W^*)] \\ & + E_{W^*}[\{g(W^*+1) - g(\lceil \frac{W^*}{2} \rceil)\}\phi(E[V^*], W^*)] \end{aligned}$$

Comparing the above expression with (5.42), we have

$$\begin{aligned} E_{W^*}[\{g(W^*+1) - g(\lceil \frac{W^*}{2} \rceil)\}\{f_{eff}(E[W^*], W^*) \\ - f_{eff}(E[V^*], W^*)\}] = E_{W^*}[\{g(W^*+1) - g(\lceil \frac{W^*}{2} \rceil)\} \\ \times \phi(E[V^*], W^*)]. \end{aligned}$$

Now, observing that $\phi()$ is a non-negative function (see 5.35) and that $g()$ can be any arbitrary bounded measurable function, the above relation implies that $f_{eff}(E[W^*], W^*) \geq f_{eff}(E[V^*], W^*)$. Also, since by Assumption 13, $f_{eff}(x, y)$ is increasing in variable x, it follows that $E[W^*] \geq E[V^*]$.

We are now done. ∎

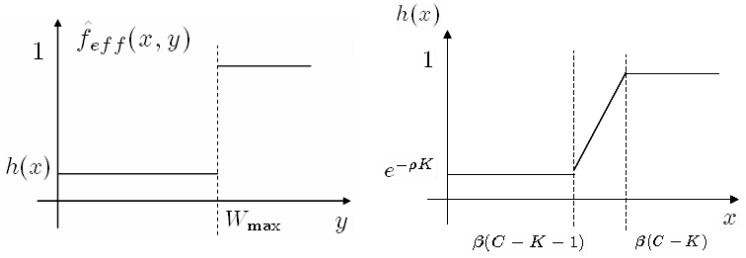

Figure 5.1: Effective marking function $\hat{f}_{eff}$.

In the following, we will use the abbreviated notation $\beta \stackrel{\Delta}{=} \alpha^{-1} E[1/P_1]^{-1}$ and

$$\rho \stackrel{\Delta}{=} \min_k D(p_k/\alpha || p_k) K.$$

Consequently, from Lemma 27, we may write $\epsilon_K = e^{-\rho K}$.

Let us now define a marking function $f(x,y)$ as follows

$$f(x,y) \stackrel{\Delta}{=} \begin{cases} h_0(x) & \text{if } 1 \le y < W_{\max} \\ 1 & \text{if } y \ge W_{\max} \end{cases} \tag{5.44}$$

where

$$h_0(x) \stackrel{\Delta}{=} \begin{cases} 0 & \text{if } x \le \beta(C-K-1) \\ x\frac{1}{\beta}(C-K) & \text{if } \beta(C-K-1) < x \\ +1-(C-K) & \le \beta(C-K) \\ 1 & \text{otherwise} \end{cases}$$

Correspondingly, from (5.36) the effective marking function $\hat{f}_{eff}(x,y)$ (see Figure 5.1) can be defined as

$$\hat{f}_{eff}(x,y) \stackrel{\Delta}{=} \begin{cases} h(x) & \text{if } 1 \le y < W_{\max} \\ 1 & \text{if } y \ge W_{\max} \end{cases} \tag{5.45}$$

where

$$h(x) \stackrel{\Delta}{=} \begin{cases} \epsilon_K & \text{if } x \le \beta(C-K-1) \\ x\frac{1-\epsilon_K}{\beta}(C-K) & \text{if } \beta(C-K-1) < x \\ +1-(1-\epsilon_K)(C-K) & \le \beta(C-K) \\ 1 & \text{otherwise} \end{cases}$$

Note that the scaling in the support of the first parameter in the above expression is due to the scaling in the first parameter between the marking function $f(\tilde{w}_{tot}, \cdot)$ and the effective marking function $f_{eff}(E[W], \cdot)$ in (5.34).

Lemma 29. *Let us consider any $\delta \in (0,1)$. If $W_{\max} = C$ and*

$$K = \frac{\log(C/2)}{\rho}$$

under the effective marking function (5.45), the ergodic distribution V^* corresponding to the recursion in (5.39) satisfies

$$E[V^*] \geq [\beta(C-K-1)]\,(1-2e^{-1}-2\delta). \tag{5.46}$$

for all $C > C_\delta$.

Proof:

Setting $g'(x) = x$ in (5.43), we have

$$E_{V^*}[1] \quad = \quad E_{V^*}[\lfloor \frac{V^*}{2} \rfloor \hat{f}_{eff}(E[V^*], V^*)] \tag{5.47}$$

Accordingly, if $E[V^*] < \beta(C-K-1)$, we may write (5.47) as follows,

$$\begin{aligned}
1 &= E_{V^*}[\lfloor \frac{V^*}{2} \rfloor \hat{f}_{eff}(E[V^*], V^*)] \\
&= \sum_{v=1}^{W_{\max}-1} \lfloor \frac{v}{2} \rfloor h(E[V^*])\mathbb{P}(V^* = v) + \lfloor \frac{W_{\max}}{2} \rfloor \mathbb{P}(v = W_{\max}) \\
&\leq \sum_{v=1}^{W_{\max}} \lfloor \frac{v}{2} \rfloor h(E[V^*])\mathbb{P}(V^* = v) + \lfloor \frac{W_{\max}}{2} \rfloor \mathbb{P}(v = W_{\max}) \\
&= E[\lfloor \frac{V^*}{2} \rfloor] h(E[V^*]) + \lfloor \frac{W_{\max}}{2} \rfloor \mathbb{P}(v = W_{\max}).
\end{aligned} \tag{5.48}$$

We will now bound the value of $\mathbb{P}(v = W_{\max})$.

Consider the Markov Chain $\{V(t)\}$, governed by equation (5.39). In this Markov Chain, let us define the event $\mathcal{S}_i(t)$ to denote that $V(t)$ is in state $\{V(t) = i\}$, at time t, for $i = 1, 2, \ldots, W_{\max}$.

Since $V^* \leq W_{\max}$ always, we observe from the dynamics of (5.39), that each state $\{V(t) = j\}$, $j > \lceil W_{\max}/2 \rceil$, can only be reached via state $\{V(t-1) = j-1\}$; in other words, there is only one inward state transition to these states, and that transition is from the smaller window size. Also, since the state transition probabilities are invariant with time, let us define $\mathbb{P}_{j,i} \stackrel{\Delta}{=} \mathbb{P}(V(t) = j | V(t-1) = i)$.

Hence, for each $j > \lceil W_{\max}/2 \rceil$, we may write $\mathbb{P}(\mathcal{S}_j(t)) = \mathbb{P}(\mathcal{S}_{j-1}(t-1))\mathbb{P}_{j,j-1}$.

Thus,

$$\begin{aligned} & \mathbb{P}(\mathcal{S}_{W_{\max}}(t + \lceil W_{\max}/2 \rceil - 1)) \\ = \; & \mathbb{P}(\mathcal{S}_{\lceil W_{\max}/2 \rceil - 1}(t)) \prod_{j=\lceil W_{\max}/2 \rceil}^{W_{\max}} \mathbb{P}_{j,j-1}. \end{aligned}$$

Now, since $\hat{f}_{eff}(E[V^*], v) = h(E[V^*])$ for all $v < W_{\max}$, $\mathbb{P}_{j,j-1} = (1 - h(E[V^*]))$. Further, we use the well known result [59] that the stationary probability of a Markov process at state $\mathcal{S}_i$ satisfies

$$\mathbb{P}(\mathcal{S}_i) = E[T_i]^{-1}$$

where T_i is the sojourn time of the Markov process starting from state $\{V = i\}$. Observe that if, starting from state $\{V = \lceil W_{\max}/2 - 1 \rceil\}$ the window size increases by $k \geq 0$ steps before halving, the total number of elapsed time-steps before V^* enters state $\{V = W_{\max}/2 - 1\}$ again, is at least $k + \lceil W_{\max}/2 \rceil - \lceil (\lceil W_{\max}/2 \rceil + k)/2 \rceil$. Minimizing over all k, we find that the minimum size of the sojourn time starting from $\{V = \lceil W_{\max}/2 \rceil\}$ is at least $0.5 * (\lceil W_{\max}/2 \rceil)$. Therefore, it follows that $E[T_i] > 0.5 * (\lceil W_{\max}/2 \rceil)$; and consequently that $\mathbb{P}(\mathcal{S}_{\lceil W_{\max}/2 \rceil - 1}) < 2/\lceil W_{\max}/2 \rceil$.

Thus we can write,

$$\mathbb{P}(\mathcal{S}_{W_{\max}}) \leq \frac{2}{\lceil W_{\max}/2 \rceil} (1 - h(E[V^*]))^{\lfloor \frac{W_{\max}}{2} \rfloor}$$

Substituting the above bound in (5.48),

$$1 \leq E[\lfloor \frac{V^*}{2} \rfloor] h(E[V^*]) + 2(1 - h(E[V^*]))^{\lfloor \frac{W_{\max}}{2} \rfloor}. \tag{5.49}$$

Let us now consider three possible cases: **(i)** $E[V^*] < \beta(C - K - 1)$, **(ii)** $E[V^*] > \beta(C - K)$ and **(iii)** $\beta(C - K - 1) \leq E[V^*] \leq \beta(C - K)$. Since the Markov chain $\{V(t)\}$ reaches a unique finite stationary distribution, the value of $E[V^*]$ must lie in one of the above three regions.

Observe that if $\bar{v} \triangleq E[V^*]$ exists in either the region corresponding to case(ii) or case(iii), it automatically implies that $\bar{v} = E[V^*] > \beta(C - K - 1)(1 - 2(e^{-1} + \delta))$. Hence, we only need to show that if $\bar{v}$ is in the region corresponding to case (i), that it still satisfies the bound in (5.46).

Null Hypothesis: Let us consider the null hypothesis that $E[V^*] < \beta(C - K - 1)(1 - 2(e^{-1} + \delta))$.

If $E[V^*] < \beta(C-K-1)$, it is clear from the defintion of the marking function that $h(E[V^*]) = \epsilon_K$.

Then, using the abbreviated notation $\bar{v} = E[V^*]$ and noting that $W_{\max} = C$, we can write (5.49) as

$$1 \leq \frac{\epsilon_K \bar{v}}{2} + 2(1-\epsilon_K)^{\lfloor C/2 \rfloor}. \tag{5.50}$$

Now, since $\epsilon_K = e^{-K\rho}$, and $K = \frac{\log(C/2)}{\rho}$, we have $\epsilon_K = \frac{2}{C}$.

Then, using (5.50) and noting that $\lfloor x \rfloor \leq x$, $\bar{v}$ must satisfy

$$1 \leq \frac{2}{C}\frac{\bar{v}}{2} + 2(1-\frac{2}{C})^{\lfloor C/2 \rfloor}.$$

As $C \to \infty$, the second term converges to e^{-1}. Hence, for any $\delta > 0$, we can find a $C_{\delta,1}$ large enough so that

$$1 \leq \frac{2}{C}\frac{\bar{v}}{2} + 2(e^{-1}+\delta).$$

Rearranging terms, we arrive at,

$$\bar{v} \geq (1-2(e^{-1}+\delta))C.$$

Since $\beta < 1$ and $K > 0$, this immediately leads to a contradiction to the null hypothesis of Case (i) that $\bar{v} < \beta(C-K-1)(1-2(e^{-1}+\delta))$. Hence, by proof by contradiction, we are done. ∎

Remark 17. *Let us briefly consider the hypothesis that Case (ii) occurs, i.e.* $\bar{v} = E[V^*] > \beta(C-K)$.

In this range of values of $E[V^*]$, $h(\bar{v}) = 1$. *However,* $h(\bar{v}) = 1$ *implies that* $\hat{M}(t) = 1$ *almost surely at all times. Hence, the window size process* $V(t) \leq 2$ *almost surely. This contradicts the hypothesis that* $E[V^*] > \beta(C-K)$, *thereby ruling out this case.*

Theorem 9. *For the system constrained by system parameters* $K, W_{\max}$ *defined in (29), for any* $\delta > 0$, *there exists a* C_δ *such that for all* $C > C_\delta$, $E[W^*] \geq \left[\alpha^{-1}E[1/P_1](C-K-1)\right](1-2e^{-1}-2\delta)$.

Proof: From Lemmas 29 and 27, $E[V^*] \geq \left[\alpha^{-1}\mathbb{E}[1/P_1](C-K-1)\right](1-2e^{-1}-2\delta)$. The result now follows from the bound $E[V^*] \leq E[W^*]$ in Lemma 28. ∎

We remark that $E[W^*] \geq \left[\alpha^{-1}E[1/P_1](C-K-1)\right](1-2e^{-1}-2\delta)$, implies that a mean window size linear in C is possible using the elementary marking function f as defined in (5.45). Moreover, the mean window size is lower bounded by approximately $(1-2e^{-1})$ times the ergodic per flow capacity for each user.

5.5 Proofs

5.5.1 Proof of Lemma 23

From the window evolution equation (5.17),

$$W^1_{(N)}(t+1) = \begin{cases} W^1_{(N)}(t)+1 & \text{if} \quad M^1_{(N)}(t)=0 \\ \lceil \frac{W^1_{(N)}(t)}{2} \rceil & \text{if} \quad M^1_{(N)}(t)=1 \end{cases}$$

It follows from (5.17) that

$$\begin{aligned} g(W^1_{(N)}(t+1)) &= (1-M^1_{(N)}(t))g(W^1_{(N)}(t)+1) \\ &\quad + M^1_{(N)}(t)g(\lceil \frac{W^1_{(N)}(t)}{2} \rceil). \end{aligned} \tag{5.51}$$

Consider

$$E\left[M^1_{(N)}(t)|\mathcal{F}_t\right] = 1 - E\left[A^1_{(N)}(t)|\mathcal{F}_t\right] E\left[B^1_{(N)}(t)|\mathcal{F}_t\right].$$

We know, from [A: t] and [B: t], the tuple

$$(N^{-1}\tilde{W}_{(N)}(t), W^1_{(N)}(t)) \Rightarrow_N (\tilde{w}_{tot}(t), W(t)). \tag{5.52}$$

Let us define the random variables

$$\begin{aligned} Z^1_{(N)}(t) \stackrel{\Delta}{=} E[A^1_{(N)}(t)|\mathcal{F}_t] &= 1 - f^{(N)}(\tilde{W}_{(N)}(t), W^1_{(N)}(t)) \\ &= 1 - f(N^{-1}\tilde{W}_{(N)}(t), W^1_{(N)}(t)), \end{aligned} \tag{5.53}$$

and

$$Y^1_N(t) \stackrel{\Delta}{=} E[B^1_{(N)}(t)|\mathcal{F}_t]. \tag{5.54}$$

Using (5.52) and considering the continuity of the function in (5.53), the continuous mapping theorem yields

$$(N^{-1}\tilde{W}_{(N)}(t), W^1_{(N)}(t), Z^1_{(N)}(t)) \Rightarrow_N (\tilde{w}_{tot}(t), W(t), Z^1(t)), \tag{5.55}$$

where $Z^1(t) = 1 - f(\tilde{w}_{tot}(t), W(t))$.

We will next show that

$$(W^1_{(N)}(t), Y^1_{(N)}(t)) \Rightarrow_N (W(t), Y^1(t)). \tag{5.56}$$

Since $W^1_{(N)}(t) \leq W_{\max}$ and $P^1_t > 0$, the total number of packets (data + auxiliary), $G(P^1_t, W^1_{(N)}(t))$, transmitted by the router R is a finite integer.

Then, following the definition of the empirical probability of success over the wireless channel $\hat{P}^1_t$ from (5.13), we find that $\hat{P}^1_t$ is a rational number such that $\hat{P}^1_t G(P^1_t, W^1_{(N)}(t)) \in \mathbb{N}$.

Hence, we can write,

$$\begin{aligned} & Y^1_{(N)}(t) \\ = \; & E[B^1_{(N)}(t)|\mathcal{F}_t] \\ = & \sum_{p=1}^{G(P^1_t, W^1_{(N)}(t))} \mathbf{1}_{p>P^1_t} \binom{G(P^1_t, W^1_{(N)}(t))}{p} (P^1_t)^p (1-P^1_t)^{G(P^1_t, W^1_{(N)}(t))-p}. \end{aligned}$$

Since $G(P^1_t, W^1_{(N)}(t))$ is finite, the above sum is a finite binomial sum. Further, recalling the definition of $G(P^1_t, W^1_{(N)}(t))$ from (5.5) each term in the sum above is a continuous function of $W^1_N(t)$. From the above two statements, it follows that $Y^1_{(N)}(t)$ is continuous in $W^1_N(t)$, i.e. there exists a continuous function $g' : \{0,1\} \to \mathbb{R}$. Accordingly, we can use the continous mapping theorem again to arrive at (5.56).

Then for any arbitrary fixed mapping $g : \mathbb{N} \to \mathbb{R}$,

$$\begin{aligned} & E[g(W^1_{(N)}(t+1)|\mathcal{F}_t] \\ = \; & E[1 - M^1_{(N)}(t) g(W^1_{(N)}(t)+1)|\mathcal{F}_t] \\ & + E[M^1_{(N)}(t).g(\lceil \frac{W^1_{(N)}(t)}{2} \rceil)|\mathcal{F}_t] \\ = \; & \Big[E[A^1_{(N)}(t)|\mathcal{F}_t] E[B^1_{(N)}(t)|\mathcal{F}_t]\Big] g(W^1_{(N)}(t)+1) \\ & + \Big[1 - E[A^1_{(N)}(t)|\mathcal{F}_t] E[B^1_{(N)}(t)|\mathcal{F}_t]\Big] g(\lceil \frac{W^1_{(N)}(t)}{2} \rceil) \\ = \; & Z^1_{(N)}(t) Y^1_{(N)}(t) g(W^1_{(N)}(t)+1) \\ & + \Big\{1 - Z^1_{(N)}(t) Y^1_{(N)}(t)\Big\} g(\lceil \frac{W^1_{(N)}(t)}{2} \rceil) \end{aligned} \tag{5.57}$$

From (5.55) and (5.56),

$$\begin{aligned} & (N^{-1}\tilde{W}_{(N)}(t), W^1_{(N)}(t), Z^1_{(N)}(t), Y^1_{(N)}(t)) \\ & \qquad \Rightarrow_N (\tilde{W}_{tot}(t), W(t), Z^1(t), Y^1(t)), \end{aligned} \tag{5.58}$$

and the function

$$\phi_g(w,z,y) \stackrel{\Delta}{=} zyg(w+1) + (1-zy)g(\lceil w \rceil) \tag{5.59}$$

is continuous on $\mathbb{N} \times \mathbb{R}^2$. It follows from the continuous mapping theorem again, that

$$\phi_g(W^1_{(N)}(t), Z^1_{(N)}(t), Y^1_{(N)}(t)) \Rightarrow_N \phi_g(W(t), Z^1(t), Y^1(t)). \tag{5.60}$$

Hence,

$$\begin{aligned}
& \lim_{N\to\infty} E[g(W^1_{(N)}(t+1))] \\
= & \lim_{N\to\infty} E[E[g(W^1_{(N)}(t+1))|\mathcal{F}_t]] \\
= & \lim_{N\to\infty} E[\phi(W^1_{(N)}(t), Z^1_{(N)}(t), Y^1_{(N)}(t))] \\
= & E[\phi_g(W^1(t), Z^1(t), Y^1(t))] \\
= & E[g(W(t+1))].
\end{aligned}$$

Since $g()$ is arbitrary, this implies that $W^1_{(N)}(t+1) \Rightarrow_N W(t+1)$.

5.5.2 Proof of Lemma 24

This result is analogous to the corresponding asymptotic independence result in [88]; we present it here only for sake of completeness.

Consider any finite subset $\mathcal{I} \subseteq \mathbb{N}$. For any flow $k \in \mathcal{I}$, the random variables V^k_t, P^k_t and the channel error event (corresponding to H_j) evolve independently of the filtration $\mathcal{F}_t$. Consequently, looking at the expression for window size evolution in (5.51), we see that $W^k_{(N)}(t+1)$ are mutually independent when conditioned on $\mathcal{F}_t$. As a result, for any set of arbitrary functions $g_k : \mathbb{N} \to \mathbb{R}$, $k \in \mathcal{I}$ we get

$$\begin{aligned}
& E[\prod_{k\in\mathcal{I}\cap[1,N]} g_k(W^k_{(N)}(t+1))|\mathcal{F}_t] \\
= & \prod_{k\in\mathcal{I}\cap[1,N]} E[g_k(W^k_{(N)}(t+1))|\mathcal{F}_t] \\
= & \prod_{k\in\mathcal{I}\cap[1,N]} E[\phi_{g_k}(W^k_{(N)}(t), Z^k_{(N)}(t), Y^k_{(N)}(t))]
\end{aligned}$$

where the last step follows from (5.57) and (5.59). Since by our hypothesis **[C:t]** holds,

$$\{W^k_{(N)}(t) : k \in \mathcal{I} \cap [1,N]\} \Rightarrow_N \{W^k(t) : k \in \mathcal{I} \cap [1,N]\}$$

where each of the limiting distributions $W^k(t) \sim W(t)$ are i.i.d. Further, since **[A:t]** holds, we can use (5.55) and (5.56) to arrive at the analogous expression for (5.60)

$$\begin{aligned}&\{\phi_{g_k}(W^k_{(N)}(t), Z^k_{(N)}(t), Y^k_{(N)}(t)) : k \in \mathcal{I} \cap [1, N]\} \Rightarrow_N \\ &\{\phi_{g_k}(W^k(t), Z^k(t), Y^k(t)) : k \in \mathcal{I}\}. \end{aligned} \tag{5.61}$$

Note that in the above expression the bounding set $[1, N]$ in the left hand side converges to $\mathbb{N}$ as $N \to \infty$.

Now since $\mathcal{I}$ is a finite subset, we can use the bounded convergence theorem to write that

$$\begin{aligned}
& \lim_{N\to\infty} E[\prod_{k\in\mathcal{I}\cap[1,N]} g_k(W^k_{(N)}(t+1))] \\
& \lim_{N\to\infty} E[E[\prod_{k\in\mathcal{I}\cap[1,N]} g_k(W^k_{(N)}(t+1))|\mathcal{F}_t]] \\
= & \lim_{N\to\infty} E[\prod_{k\in\mathcal{I}\cap[1,N]} \phi_{g_k}(W^k_{(N)}(t), Z^k_N(t), Y^k_N(t))] \\
= & E[\prod_{k\in\mathcal{I}} \phi_{g_k}(W^k(t), Z^k(t), Y^k(t))] \\
= & \prod_{k\in\mathcal{I}} E[\phi_{g_k}(W^k(t), Z^k(t), Y^k(t))] \qquad (5.62) \\
= & \prod_{k\in\mathcal{I}} E[E[g_k(W^k(t+1))|\mathcal{F}_t]] \\
= & \prod_{k\in\mathcal{I}} E[g_k(W^k(t+1))].
\end{aligned}$$

where (5.62) follows since $\{W^k(t), Z^k(t), Y^k(t)\}$ are mutually i.i.d. among any $k \in \mathcal{I}$.

Now, since the $g_k()$ are any arbitrary mappings, for any $a_k \in \mathbb{N}$, $0 < a_k < W_{\max}$, we can substitute $g_k(w) = \mathbf{1}_{\{w=a_k\}}(w)$ to arrive at the result in (5.21) for time-slot $t + 1$. We are now done.

Chapter 6

Conclusion

This thesis has focused on the examination of how Random Linear Coding (i.e. randomized network coding) may provide performance gains in next generation networks. To study this, we have considered two existing techniques of coding across packets in a network: network coding at every intermediate nodes, and network coding only at source nodes.

6.1 Coding at source nodes

In Chapter 2, we have presented a cost splitting rule at each link for the min-cost problem using network coding and have demonstrated that under this rule, the user-equilibrium (assuming selfish nodes) is the same as the min-cost solution subject to certain conditions on edge cost functions. Based on this, we have provided two selfish min-cost routing algorithms UESSM and LDSRA which gave desired simulation performance. Additionally, we prove that UESSM converges to the min-cost solution for any network topology.

Next, in Chapter 3, we presented the model of a Broadcast and Additive Erasure network (BAIN), which is an abstraction of the broadcast and interference properties of the wireless network. To find the upper bound on unicast capacity, we present a transformation from a BAIN to a Broadcast erasure network studied by Gowaikar et. al. [16]. For the lower bound (achievability), we consider random linear coding at every intermediate node and present a graph transformation and a sample path coupling to map the flow of innovations over the BAIN to a corresponding flow of innovations in a wireline network. For these networks, we proved that Random Linear coding over all previously received packets at intermediate nodes can asymptotically achieve unicast capacity; if $\mathbb{F}_q$ is the finite field under consideration, we show that the gap between the upper and lower bounds is $O(1/q)$.

6.1.1 Future directions

A natural problem for future study is to see if we can use the graph transformation and sample path coupling techniques in Chapter 3 to obtain capacity results for multicast as well. Thereafter, it would be interesting to understand if we can provide a "selfish network coding min-cost multicast formulation for wireless networks, thereby extending the work in Chapter 2 to the wireless case as well. From the information theoretic standpoint, it is of interest to determine unicast capacity, under more general network models. Avestimehr, Diggavi and Tse [94] have recently presented arguments where a Gaussian wireless broadcast and additive network can be modelled as a deterministic network with finite field addition. In their work, they model the Gaussian channel as a bit pipe where bits are at different signal levels (from most significant to least) and that the mean behaviour of the channel is seen at all times. The most significant bits are deterministically received by the receiver and the less significant bits, which are below the noise floor are not received at all. This is similarly extended to the broadcast and MAC channels. This work motivates us to explore if we can similarly connect the uniform finite field assumption in our model to the Gaussian noise scenario.

Moreover, the model under consideration in Chapter 3 assumes that the inter-arrival times between innovations at the source node are iid and the packet erasure process at each receiver is independent of other erasures in the network. It would be of interest to examine how the unicast capacity expression changes when the erasure process is non-iid.

6.2 Coding at source nodes

In Chapter 4, we have presented a technique for sharing buffer resources among many links on a single path; we call this spatial buffer multiplexing. Under mild conditions on mixing of the packet arrival processes at a link, we use many-sources large deviations techniques to demonstrate that the probability that a dropped packet will not be decoded within a finite time d scales exponentially in both d and n, where n is the number of flows sharing the link. We extend this result from a single link to a set of links on a source-destination path and show that the asymptotic probability that a packet dropped on any link in the path will not be recovered within a finite time is also exponential in d and n. We then compare packet drop probability tail distributions for network coding at the source with traditional queueing and conclude that network coding promises the same asymptotic performance with $\Theta(d)$ buffers at the source and destination, but no buffers in intermediate links. Thus spatial buffer multiplexing can greatly reduce buffer allocation requirements

without sacrificing QoS in large scale networks. We observe that the gains obtained in Chapter 4 are available only for large scale networks. In fact, we consider the opposite scenario of a large source buffer but few flows sharing a link; in that case, the statistical multiplexing gains for the many-source large deviations result vanish.

Finally, in Chapter 5, we propose TCP-NC, which uses network coding at the source, and may be used to dynamically compensate for time-varying packet loss probabilities over the wireless channel in access-networks. The wireless transmitter (the base station) adaptively shares the common channel resources among the downlink nodes by varying the number of redundant packets allocated to each flow. Further, to ensure that the TCP window sizes do not grow too large to cause congestion, the base station implements an Active Queue Management system based on a packet dropping function that takes the "total coded+uncoded flow" as its arguments. We prove that as the number of flows sharing the link tends to infinity, the window size processes at each flow converge weakly. Further, to estimate the average throughput on each flow, we show that the mean window size under the stationary distribution of the window process is $(1-2e^{-}1) \times \texttt{ergodiccapacityperflow}$. This implies that the window size scales linearlyin the per-flow capacity – significantly higher than the mean window size distribution without network coding.

6.2.1 Future directions

The polynomial-order savings in buffer resources using spatial buffer multiplexing (cf. Chapter 4) is valid only for networks where the typical path length scales polynomially in the size of the network. However, many real wireline networks, such as the Internet tend to be "small-world" type power-law graphs where the typical path length scales logarithmically in the size of the network. For these networks as well, we expect gains using spatial buffer multiplexing (coding) over static buffer multiplexing (queueing). However, these gains are unlikely to be polynomial-order and thus, considering only the exponential rate-function may be inadequate – we expect that it calls for bounds on the pre-exponent of the packet loss probability.

Similarly, the many-flows analysis in Chapter is only the first step to a complete understanding of how source coding can improve TCP performance over wireless channels. It still remains to be examined if source coding can provide performance gains when only a few flows share a wireless channel. Aside from these theoretical analyses, we believe that there is a strong case for a detailed simulation level study of performance gains of TCP-NC. The mathematical analysis in Chapter 5 only assumes the basic AIMD aspect of TCP, and

skips over the protocol specific details like time-outs, varying RTTs, etc. We believe that a detailed simulation and system study, would lead to a practical scheme for implementation of TCP-NC, and to better understand its pros and cons. Preliminary numerical simulations of the fixed point for the TCP-NC window size distribution indicate that the mean window distribution is close to the ergodic mean; this suggests that a tighter bound than $(1 - 2e^{-}1)$ may be proved for the fixed point.

Bibliography

[1] V. Anantharam and S. Verdu, "Bits through queues," *IEEE Transactions of Information Theory*, vol. 42, no. 1, pp. 4–18, 1996.

[2] R. Gallager and B. Prabhakar, "The entropies of queue arrivals and queue departures," in *Information Theory and Networking Workshop, 1999*, 1999, pp. 42–.

[3] J. Giles and B. Hajek, "An information-theoretic and game-theoretic study of timing channels," *IEEE Transactions of Information Theory*, vol. 48, no. 9, pp. 2455–2477, 2002.

[4] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," vol. 46, pp. 1204–1216, 2000.

[5] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, October 2003.

[6] S. Jaggi, P. A. Chou, and K. Jain, "Low complexity algebraic network multicast codes," in *ISIT*, Yokohama, Japan, 2003.

[7] P. S. S. Jaggi, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, "Polynomial time algorithms for multicast network code construction," *IEEE Transactions on Information Theory*, vol. 51, no. 6, June 2005.

[8] T. Ho, R. Koetter, M. Médard, D. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *Proc. 2003 International Symposium on Information Theory*. IEEE, 2003.

[9] T. Ho, M. Médard, J. Shi, M. Effros, and D. R. Karger, "On randomized network coding," in *41st Allerton Annual Conference on Communication, Control, and Computing*, Monticello, IL, October 2003.

[10] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *41st Allerton Annual Conference on Communication, Control, and Computing*, Monticello, IL, October 2003.

[11] A. R. Lehman and E. Lehman, "Complexity classification of network information flow problems," in *Proc. ACM-SIAM Symposium on Discrete Algorithms SODA-2004*, January 2004.

[12] D. S. Lun, M. Médard, T. Ho, and R. Koetter, "Network coding with a cost criterion," in *Proc. 2004 International Symposium on Information Theory and its Applications (ISITA 2004)*, October 2004.

[13] D. S. Lun, N. Ratnakar, R. Koetter, M. Médard, E. Ahmed, and H. Lee, "Achieving minimum cost multicast: A decentralized approach based on network coding," in *Proc. IEEE Infocom 2005*, March 2005.

[14] R. Johari and J. Tsitsiklis, "Efficiency loss in a network resource allocation game," *Mathematics of Operations Research*, vol. 29, no. 3, pp. 407–435, 2004.

[15] D. Acemoglu and A. Ozdaglar, "Flow control, routing and performance from service provider viewpoint," *MIT-LIDS-WP-1696 Technical Report, MIT-LIDS*, December 2003.

[16] R. Gowaikar, A. F. Dana, R. Palanki, B. Hassibi, and M. Effros, "On the capacity of wireless erasure networks," in *ISIT*, Chicago, IL, 2004.

[17] D. S. Lun, M. Médard, and M. Effros, "On coding for reliable communication over packet networks," in *Proc. 42nd Annual Allerton Conference on Communication, Control, and Computing*, September 2004.

[18] S. Deb and M. Médard, "Algebraic gossip: A network coding approach to optimal multiple rumor mongering," in *Proc. Allerton Conference on Communication, Control, and Computing*, Monticello, IL, September 2004.

[19] C. Gkantsidis and P. R. Rodriguez, "Network coding for large scale content distribution," in *Proc. INFOCOM 2005*, 2005.

[20] D. Mosk-Aoyama and D. Shah, "Information dissemination via gossip: Applications to averaging and coding," http://www.arxiv.org/, April 2005.

[21] D. S. Lun, M. Médard, R. Koetter, and M. Effros, "Further results on coding for reliable communication over packet networks," in *2005 IEEE International Symposium on Information Theory*, Sydney, 2005.

[22] M. Charikar, C. Chekuri, T. y. Cheung, Z. Dai, A. Goel, S. Guha, and M. Li, "Approximation algorithms for directed steiner problems," in *Proc. Ninth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 1998)*, 1998, pp. 192–200.

[23] L. Zosin and S. Khuller, "On directed steiner trees," in *Proc. 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2002)*, 2002, pp. 59–63.

[24] F. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *Journal of the Operations Research Society*, vol. 49, pp. 237–252, 1998.

[25] K. Jain, V. V. Vazirani, and Y. Yu, "Market equilibria for homothetic, quasi-concave utilities and economies of scale in production," in *Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA 2005)*, January 2005.

[26] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*. MIT Press, 1994.

[27] T. Roughgarden and E. Tardos, "How bad is selfish routing?" *Journal of the ACM*, vol. 49, no. 2, pp. 236–259, March 2002.

[28] C. Papadimitriou, "Algorithms, games, and the internet," in *Proc. 33rd Annual ACM Symposium on the Theory of Computing*, 2001, pp. 749–753.

[29] S. C. Dafermos and F. T. Sparrow, "The traffic assignment problem for a general network," *Journal of Research of the National Bureau of Standards, Series B*, vol. 73B, no. 2, pp. 91–118, 1969.

[30] M. Beckmann, C. B. McGuire, and C. B. Winsten, *Studies in the Economics of Transportation*. Yale University Press, 1956.

[31] T. Roughgarden, "The price of anarchy is independent of the network topology," in *Annual ACM Symposium on the Theory of Computing*, 2002, pp. 428–437.

[32] J. Correa, A. Schulz, and N. Stier-Moses, "Selfish routing in capacitated networks," *Mathematics of Operations Research*, vol. 29, no. 4, pp. 961–976, November 2004.

[33] N. Ratnakar and G. Kramer, "Separation of channel and network coding in aref networks," in *International Symposium on Information Theory*, Adelaide, Australia, 2005.

[34] S. Ray, M. Médard, and J. Abounadi, "Noise-free multiple access networks over finite fields," in *Proc. 41st Allerton Conference on Communication, Control and Computing*, Monticello, IL, 2003.

[35] ——, "Random coding in noise-free multiple access networks over finite fields," in *Proc. Communication Theory Symposium, Globecom*, Fremont, CA, 2003.

[36] B. Cohen, "Incentives build robustness in Bittorrent," in *P2P Economics Workshop*, Berkeley, CA, 2003.

[37] M. Luby, "LT codes," in *Proc. 43rd Annual IEEE Symposium on Foundations of Computer Science*, November 2002, pp. 271–280.

[38] M. Luby, M. Mitzenmacher, A. Shokrollahi, D.A. Spielman and V. Stemann, "Practical erasure resilient codes," in *Proc. 29th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 150–159, 1997.

[39] J. Byers, M. Luby, and M. Mitzenmacher, "Accessing Multiple Mirror Sites in Parallel: Using Tornado Codes to Speed Up Downloads," in *Proceedings of the 18th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '99)*, pp. 275–284, 1999.

[40] A. Shokrollahi, "Raptor codes," *IEEE Transactions of Information Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.

[41] S. Deb and R. Srikant, "Congestion control for fair resource allocation in networks with multicast flows," *IEEE/ACM Transactions on Networking*, pp. 274–285, April 2004.

[42] S. Bhadra, S. Shakkottai, and P. Gupta, "Min-cost selfish multicast with network coding," Bell Labs Technical Report, Tech. Rep., August 2005.

[43] P. Gupta and P. R. Kumar, "A system and traffic dependent adaptive routing algorithm for ad hoc networks," in *Proc. IEEE 36th Conf. on Decision and Control*, San Diego, 1997, pp. 2375–2380.

[44] P. Gupta, "Design and performance analysis of wireless networks," Ph.D. dissertation, University of Illinois at Urbana-Champaign, August 2000.

[45] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. on Information Theory*, vol. 46, no. 2, pp. 388–404, March 2000.

[46] V. Borkar and P. Kumar, "Dynamic Cesaro-Wardrop equilibration in networks," *IEEE Transactions on Automatic Control*, vol. 48, no. 3, pp. 382–396, 2003.

[47] S. Subramanian and S. Shakkottai, "Geographic routing with limited information in sensor networks," in *The Fourth International Conference on Information Processing in Sensor Networks(IPSN)*, Los Angeles, CA, April 2005.

[48] A. E. Gamal, J. Mammen, B. Prabhakar, and D. Shah, "Throughput delay trade-off in wireless networks," in *Proc. INFOCOM 2004*, 2004.

[49] M. Grossglauser and D. Tse, "Mobility increases the capacity of ad-hoc wireless networks," in *IEEE INFOCOM-2001*, Anchorage, Alaska, 2001, pp. 1360–1369.

[50] D. D. Botvich and N. G. Duffield, "Large deviations, the shape of the loss curve, and economies of scale in large multiplexers," *Queueing Systems*, vol. 20, pp. 293–320, 1995.

[51] S. Shakkottai and S. Srikant, "Many-sources delay asymptotics with applications to priority queues," *Queueing Systems Theory and Applications (QUESTA),*, vol. 39, pp. 183–200, October 2001.

[52] C. Courcoubetis and R. Weber, "Buffer overflow asymptotics for a buffer handling many traffic sources," *Journal of Applied Probability*, vol. 33, 1996.

[53] D. P. Bertsekas, *Nonlinear programming*. Belmont, MA: Athena Scientific, 1995.

[54] L. Kleinrock, *Queueing Systems*. John Wiley and Sons, 1976, vol. 2.

[55] A. Dembo and O. Zeitouni, *Large deviations techniques and applications*, 2nd ed. Springer-Verlag, 1998.

[56] A. Schwartz and A. Weiss, *Large deviations for performance analysis*. Chapman and Hall, 1995.

[57] G. de Veciana and J. Walrand, "Effective bandwidths: Call admission, traffic policing and filtering for atm networks," *Queueing Systems*, vol. 20, pp. 37–39, 1995.

[58] S. Bhadra and S. Shakkottai, "Looking at large networks: Coding vs. queueing," in *Proc. IEEE INFOCOM*, Barcelona, Spain, 2006.

[59] P. Billingsley, *Probability and Measure*, 3rd ed. Wiley, 1995.

[60] D. Aldous and J. Fill, *Reversible Markov Chains and Random Walks on Graphs*. http://www.stat.berkeley.edu/users/aldous/RWG/book.html.

[61] A. Das and R. Srikant, "Diffusion approximations for a single node accessed by congestion controlled sources."

[62] C. Gkantsidis, M. Mihail, and A. Saberi, "Conductance and congestion in power law graphs," in *Proc. ACM Sigmetrics*, 2003.

[63] ——, "Random walks in peer-to-peer networks," in *Proc. IEEE INFOCOM-2004*, 2004.

[64] B. Bollobas, *Random Graphs*, 2nd ed. Cambridge University Press, 2001.

[65] S. Kattia, D. Katabi, W. Hu, and R. Hariharan, "The importance of being opportunistic: Practical network coding for wireless environments," in *Proc. 43rd Annual Allerton Conference*, Monticelo, IL, September 2005.

[66] J. Widmer, C. Fragouli, and J.-Y. L. Boudec, "Energy-efficient broadcasting in wireless ad-hoc networks," in *Proc. Netcod 2005*, Riva del Garda, Italy, April 2005.

[67] Y. Sagduyu and A. Ephremides, "Joint scheduling and wireless network coding," in *Proc. Netcod 2005*, Riva del Garda, Italy, April 2005.

[68] L. Subramanian, S. Agarwal, J. Rexford, and R. Katz, "Characterizing the internet hierarchy from multiple vantage points," in *IEEE INFOCOM*, 2002.

[69] M. Zorzi and R. R. Rao, "Slotted aloha with capture in a mobile radio environment," in *Proceedings of the 1994 International Zurich Seminar on Digital Communications*. London, UK: Springer-Verlag, 1994, pp. 452–463.

[70] A. F. Dana, R. Gowaikar, R. Palanki, B. Hassibi, and M. Effros, "Capacity of wireless erasure networks", *IEEE Transactions on Information Theory*, vol. 52, pp. 789-804, March 2006.

[71] D. Lun, M. Médard, R. Koetter, and M. Effros, "On coding for reliable communication over packet networks," *IEEE Trans. on Info. Theory* (submitted).

[72] N. Abramson, "The Aloha system – Another alternative for computer communications," *Proc. AFIPS Conf.*, Fall, vol. 37, 1970, pp. 281-285.

[73] D. Bertsekas and R. Gallager, *Data Networks*, Prentice-Hall, 1992.

[74] S. Ghez, S. Verdu, and S. C. Schwartz, "Stability properties of slotted Aloha with multipacket reception capability," *IEEE Trans. Automatic Contr.*, vol. AC-33, no. 7, pp. 640-649, 1988.

[75] H. Chen and D. D. Yao, "*Fundamentals of Queueing Networks: Performance, Asymptotics and Optimization*," Springer, ISBN 0387951660, 2001.

[76] D. Gamarnik and A. Zeevi, "Validity of heavy traffic steady-state approximations in generalized jackson networks," *Annals of Applied Probability*, vol. 16, no. 1, pp. 56–90, 2006.

[77] G. Kesidis, J. Walrand, and C. Chang, "Effective bandwidths for multiclass Markov fluids and other ATM sources," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, 1993.

[78] C. Chang, "Stability, queue length and delay of deterministic and stochastic queueing networks," *IEEE Transactions on Automatic Control*, vol. 39, pp. 913–931, 1994.

[79] G. de Veciana and J. Walrand, "Effective bandwidths at multi-class queues," *QUESTA*, vol. 9, pp. 5–15, 1991.

[80] N. Duffield and N. O'Connell, "Large deviations and overflow probabilities for the general single server queue, with applications." *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 118, pp. 363–374, 1995.

[81] P. Glynn and W. Whitt, "Large deviations behaviour of counting processes and their inverses," *QUESTA*, vol. 17, pp. 107–128, 1994.

[82] C. Chang and J. Thomas, "Effective bandwidth in high speed digital networks," *IEEE Journal on Selected Areas in Communications*, vol. 13, pp. 1091–1100, 1995.

[83] Y. Tian, K. Xu, and N. Ansari, "TCP in wireless environments: Problems and solutions," *IEEE Radio Communications*, pp. 27–32, March 2005.

[84] T. Lakshman and U. Madhow, "The performance of TCP/IP for networks with high bandwidth delay products and random loss," *IEEE/ACM Transactions on Networking*, vol. 5, no. 3, pp. 336–350, June 1997.

[85] H. Balakrishnan, V. Padmanabhan, and R. Katz, "The effects of asymmetry on TCP performance," in *Proc. ACM/IEEE Mobicom*, September 1997, pp. 77–89.

[86] L. Benyuan, D. Goeckel, and D. Towsley, "TCP-cognizant adaptive forward error correction in wireless networks," in *GLOBECOM '02*, 2002.

[87] C. Hollot, V. Misra, D. Towsley, and W. Gong, "On designing improved controllers for AQM routers supporting TCP flows," in *Proceedings of IEEE INFOCOM*, Anchorage, AK, 2001.

[88] P. Tinnakornsrisuphap and A. Makowski, "On the behavior of ECN/RED gateways under a large number of TCP flows: Limit theorems," *Queueing Systems*, 2006.

[89] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 297–413, August 1995.

[90] F. Baccelli, D. McDonald, and J. Reynier, "A mean-field model for multiple TCP connections through a buffer implementing red," *Performance Evaluation*, vol. 49, no. 1–4, pp. 77–79, September 2002.

[91] S. Kunniyur and R. Srikant, "End-to-end congestion control schemes: utility functions, random losses and ECN marks," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 689–702, October 2003.

[92] S. Shakkottai and R. Srikant, "Mean FDE models for internet congestion control under a many-flows regime," *IEEE Transactions on Information Theory*, vol. 50, no. 6, pp. 1050–1072, June 2004.

[93] Y. Yi and S. Shakkottai, "Hop-by-hop congestion control over a wireless multi-hop network," *IEEE/ACM Transactions on Networking*, vol. 15, no. 1, pp. 133–144, 2007.

[94] S. Avestimehr, S. Diggavi and D. Tse, "A deterministic approach to wireless relay networks," *In Proc. Allerton Conference on Communication, Control, and Computing*, Illinois, September 2007.

Vita

Sandeep Bhadra graduated with a Bachelors and Masters degree, both in Electrical Engineering from the Indian Institute of Technology, Madras in 2003. In the past, he has worked at INRIA, Sophia Antipolis, France; Bell Labs Research, Lucent Technologies, USA; IBM Research, Yorktown NY and Texas Instruments, Dallas, TX, on various summer research projects. He has assisted in the instruction of graduate and undergraduate courses at the Department of ECE at UT Austin. He is currently at Texas Instruments DSPS R&D Center, Dallas, TX.

His research interests lie in the application of queueing theory, probability, graph theory, network information theory and game theory to a variety of practical network optimization problems in communication networks and supply chain management/inventory control.

Permanent address: 3636 McKinney Ave., Apt 519
Dallas, Texas 75204

This dissertation was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.